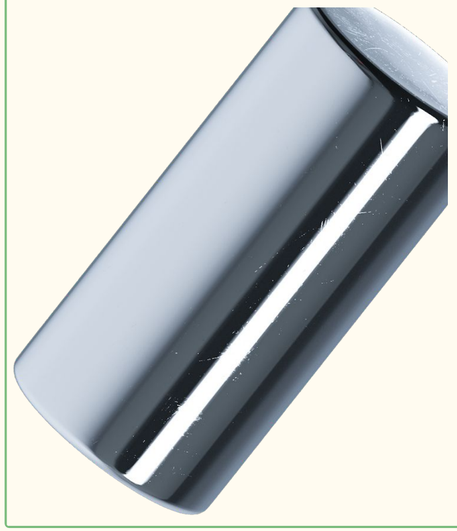
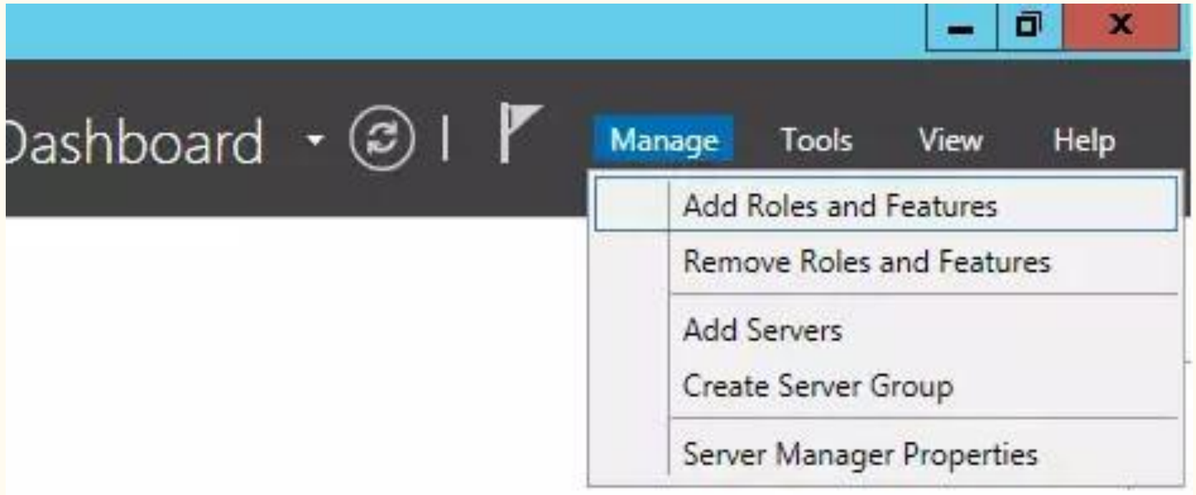
Installazione di un Server Dinamico con MySQL e Python

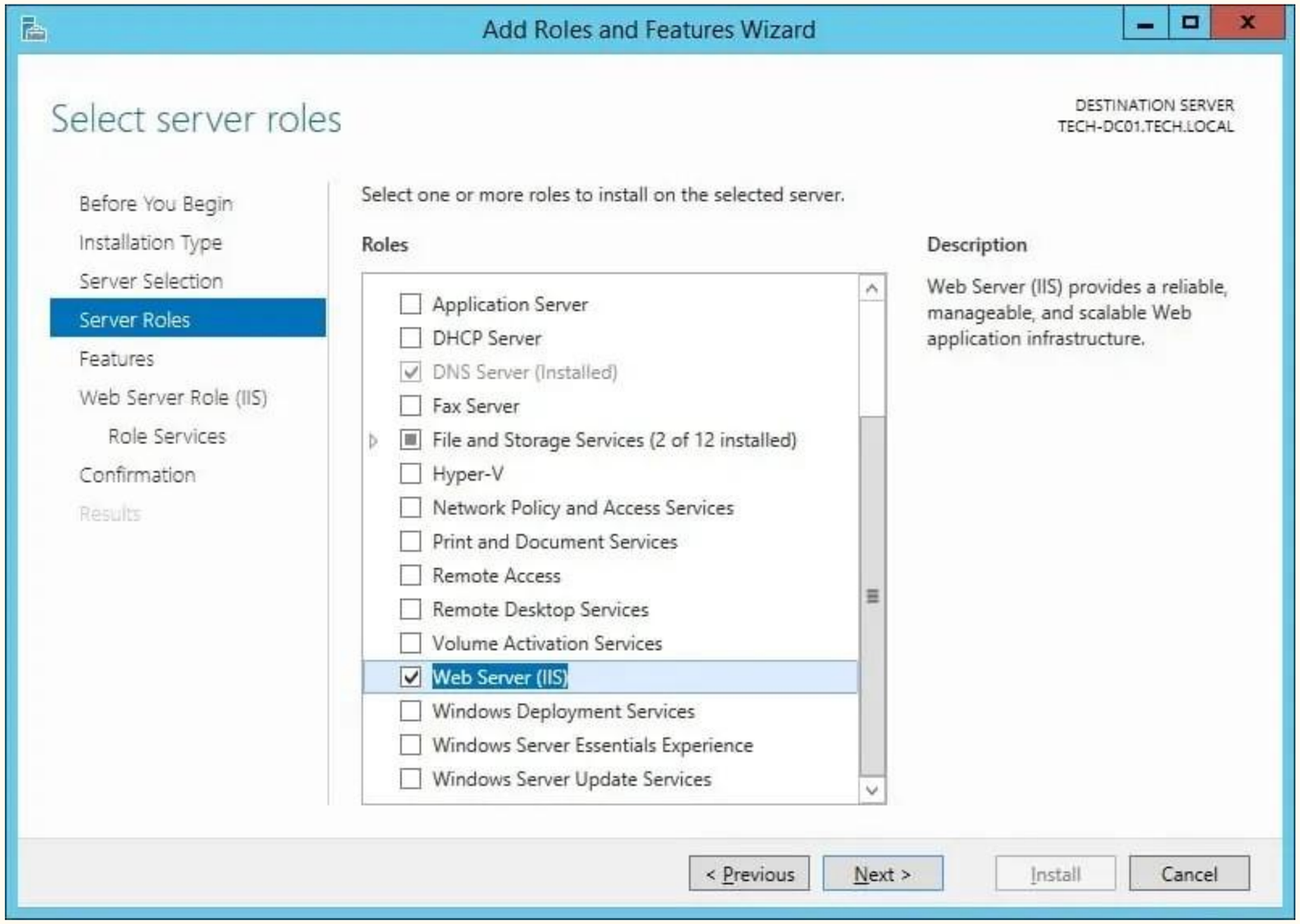


Configurazione Python Server Web

Installazione di IIS:

Aprire l'applicazione Server Manager. Accedere al menu Gestisci e fare clic su Aggiungi ruoli e funzionalità.

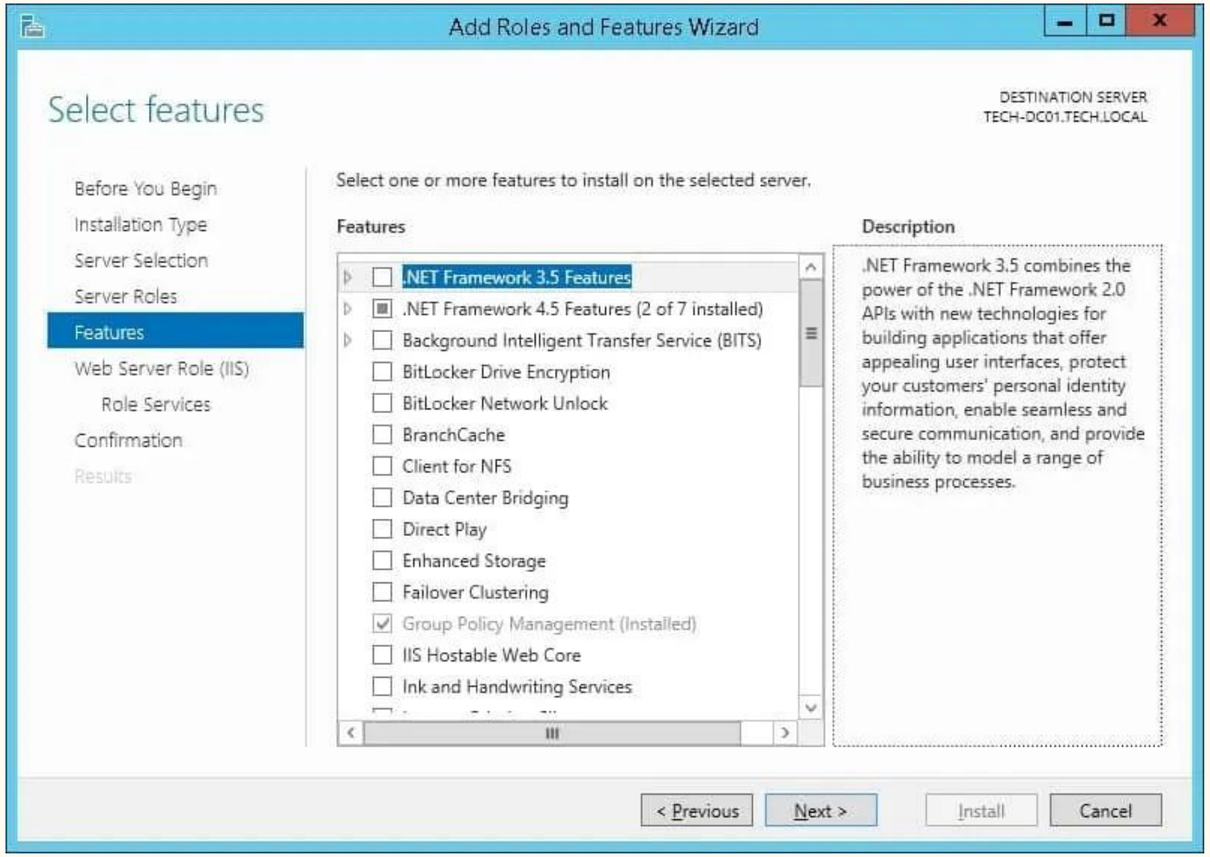




Nella schermata Ruoli server selezionare l'opzione denominata: IIS Web server Web. Fare clic sul pulsante Avanti.

Nella schermata seguente, fare clic sul pulsante Aggiungi funzionalità.

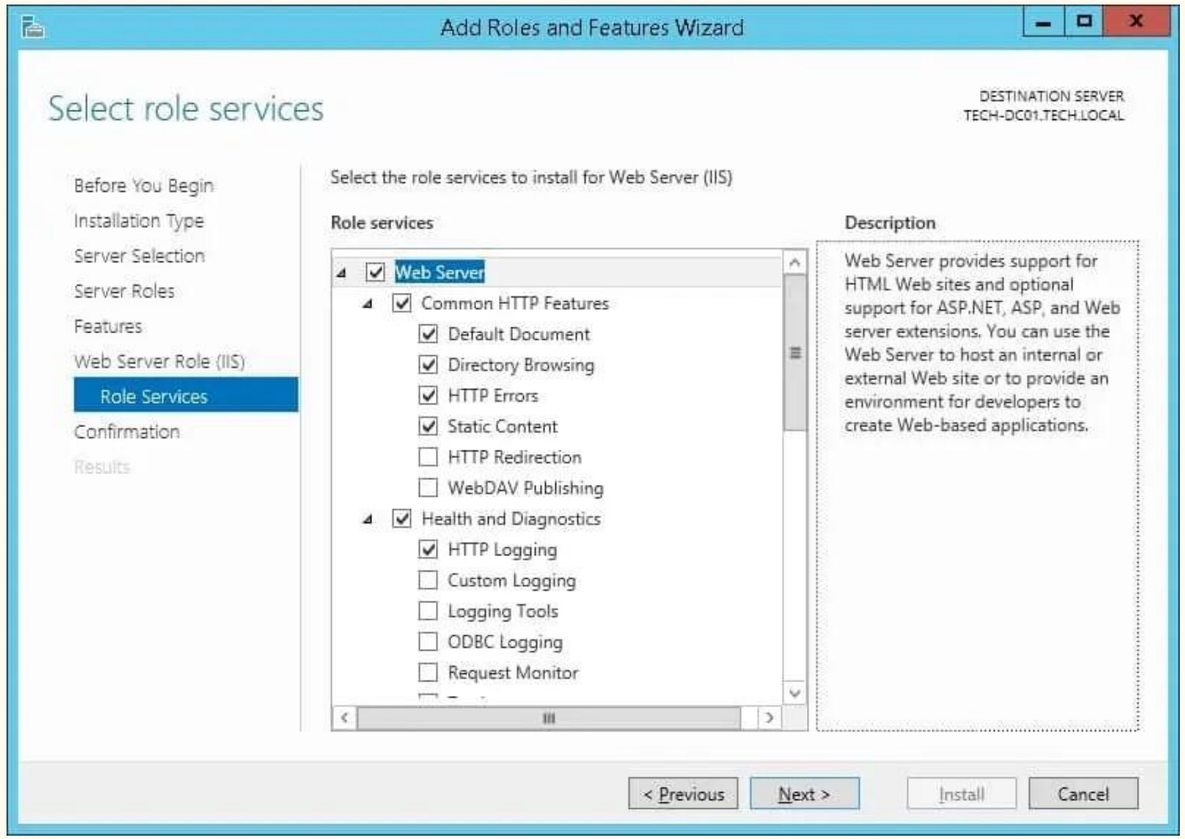
Nella schermata Funzioni, fare clic sul pulsante Avanti.

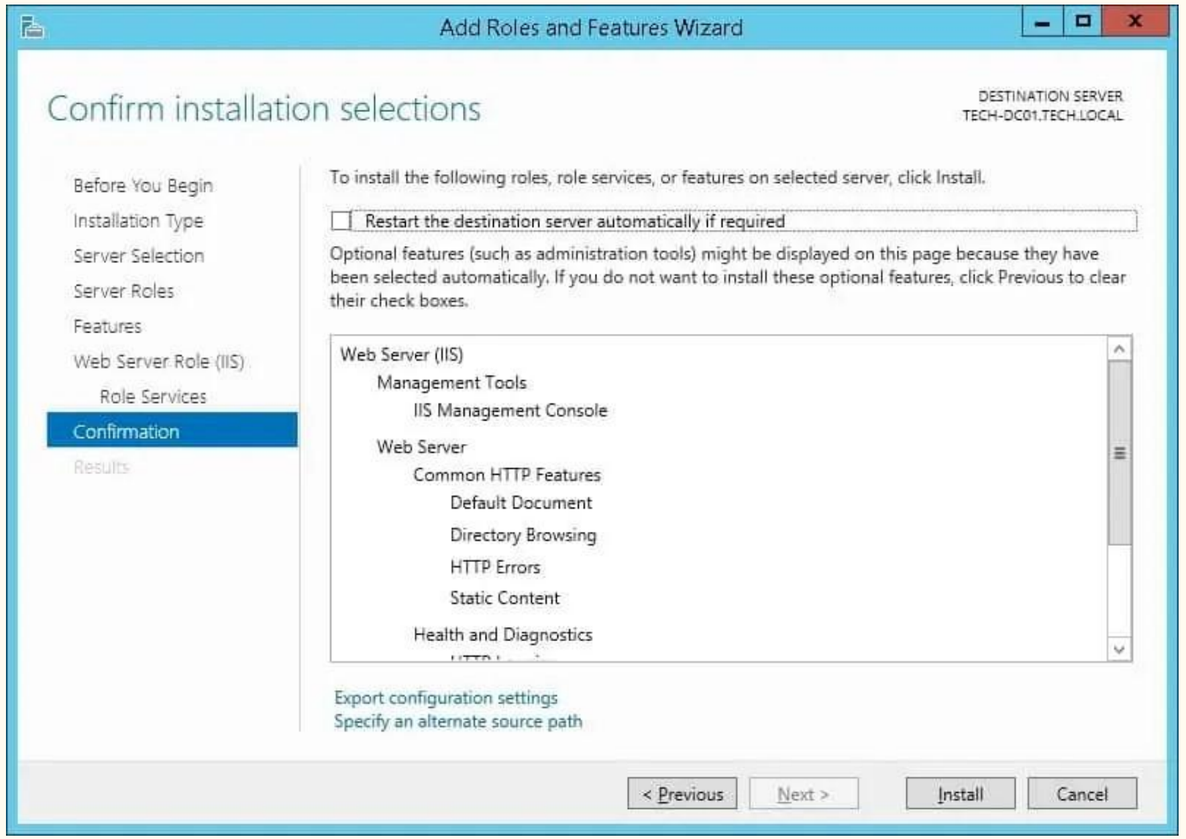


Nella schermata Servizio ruolo fare clic sul pulsante Avanti.

Nella schermata Riepilogo, fare clic sul pulsante Installa.

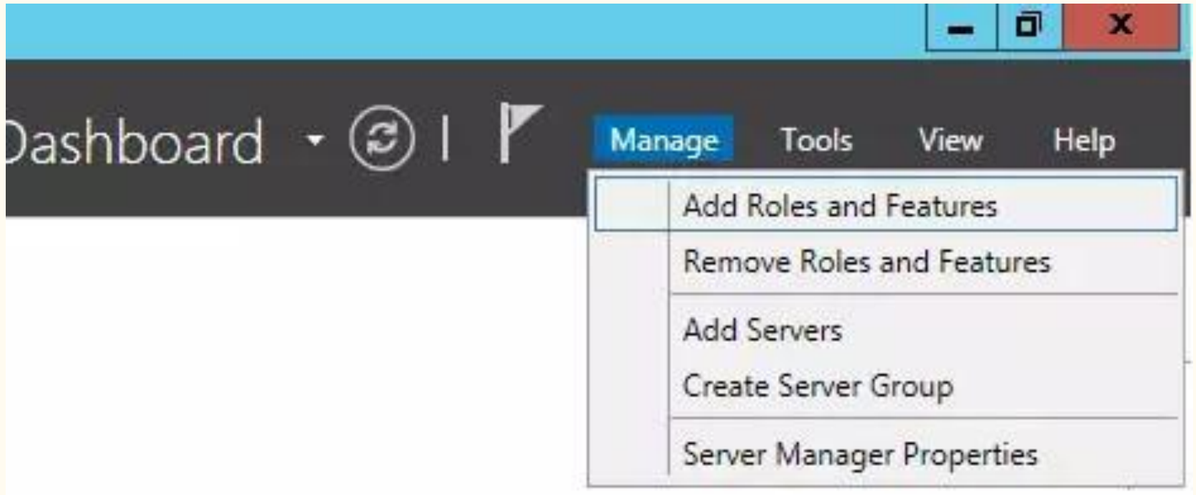
Congratulazioni! L'installazione del servizio IIS in un computer che esegue Windows è stata completata.

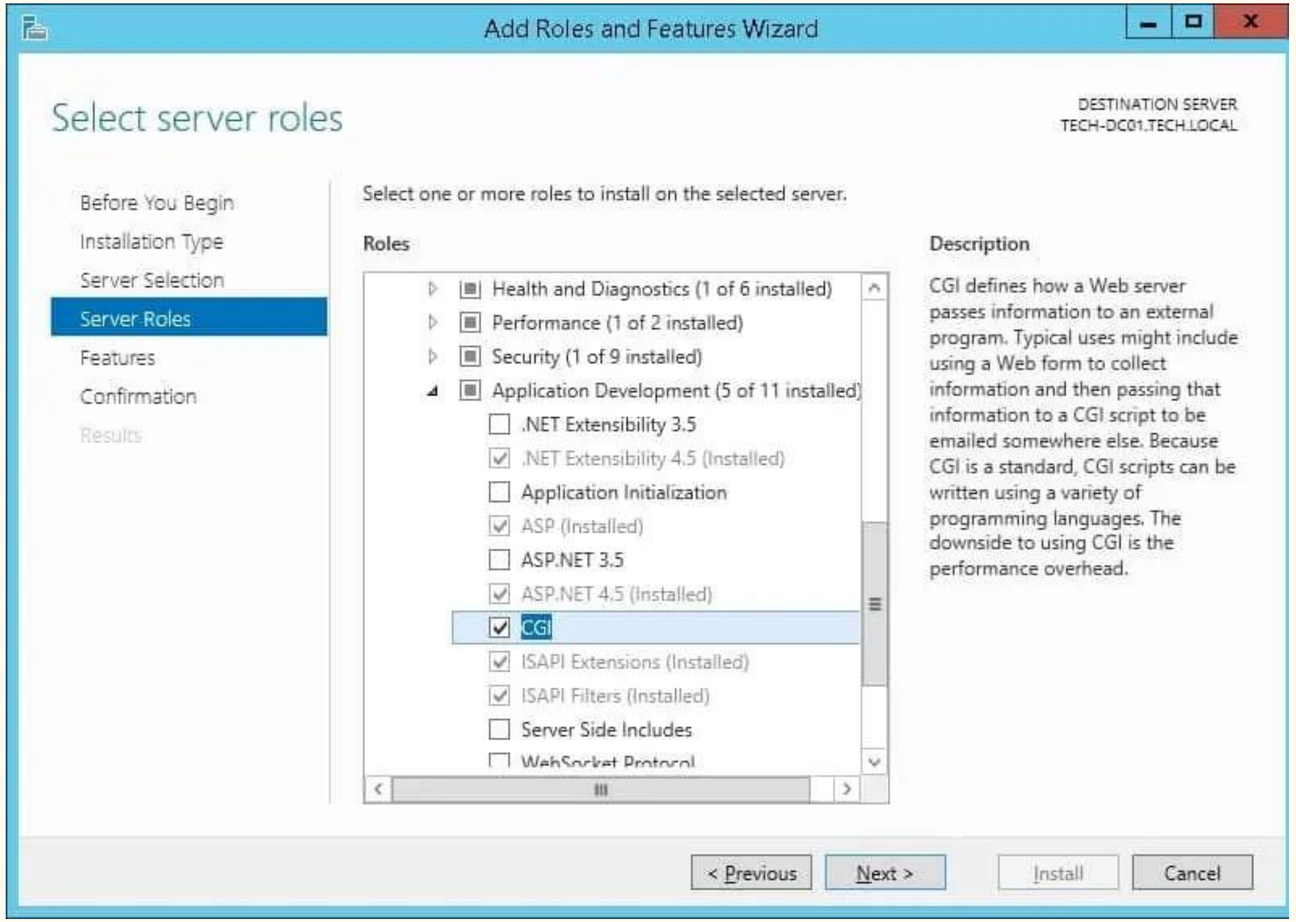




Esercitazione IIS - Abilitazione della funzionalità CGI:

Aprire l'applicazione Server Manager.
Accedere al menu Gestisci e fare clic su Aggiungi ruoli e funzionalità.

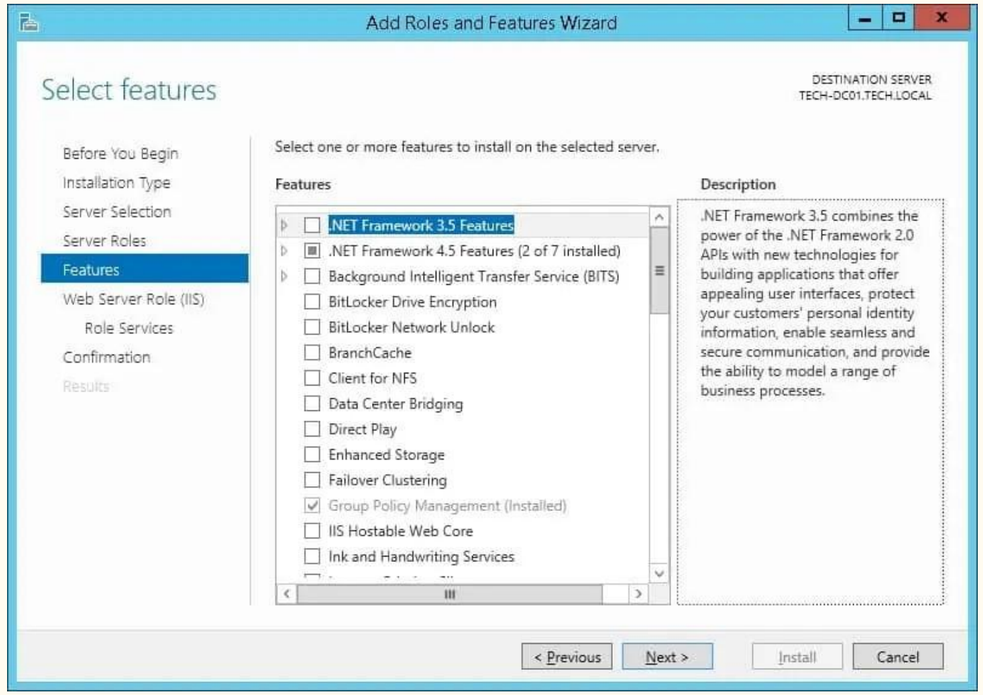




Nella schermata Ruoli server espandere la voce denominata: IIS del server Web. Accedere al menu Sviluppo applicazioni e selezionare l'opzione denominata: CGI

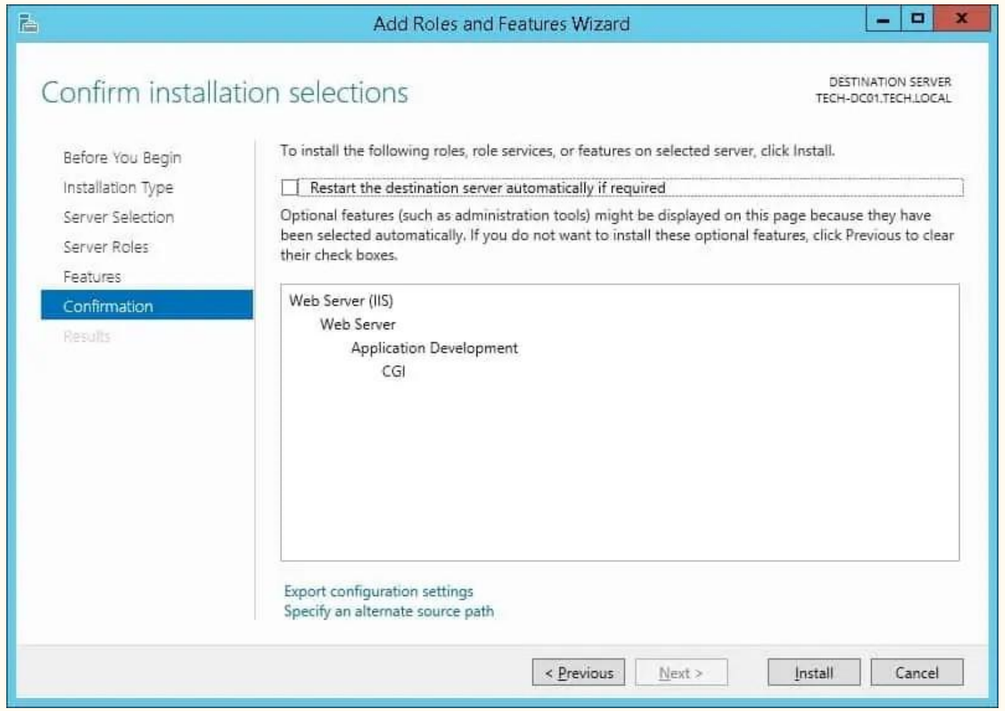
Fare clic sul pulsante Avanti.

Nella schermata Funzioni, fare clic sul pulsante Avanti.



Nella schermata Riepilogo, fare clic sul pulsante Installa.

Congratulazioni! L'installazione della funzionalità CGI in IIS è stata completata.



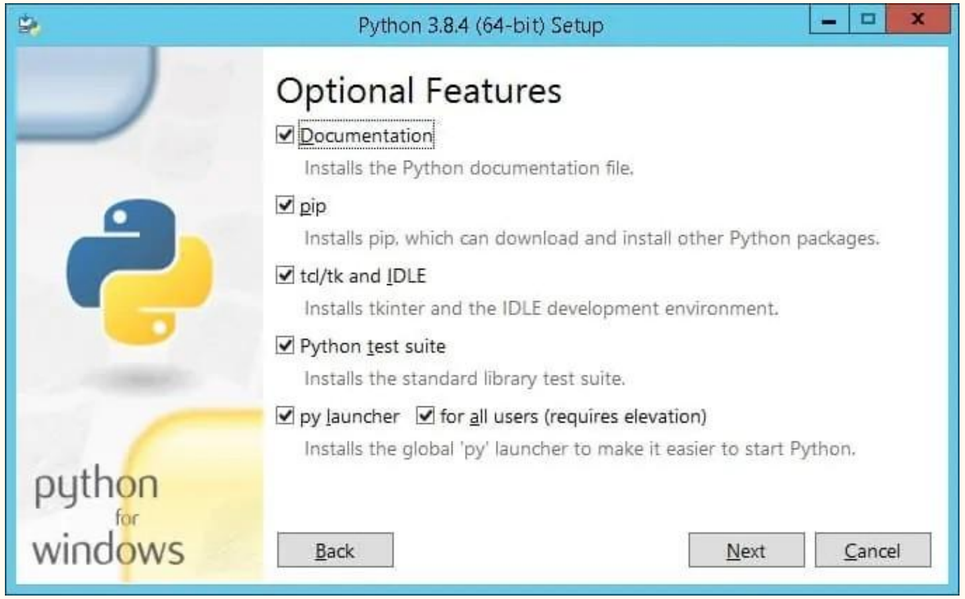
EsercitazioneWindows - Installazione Python:

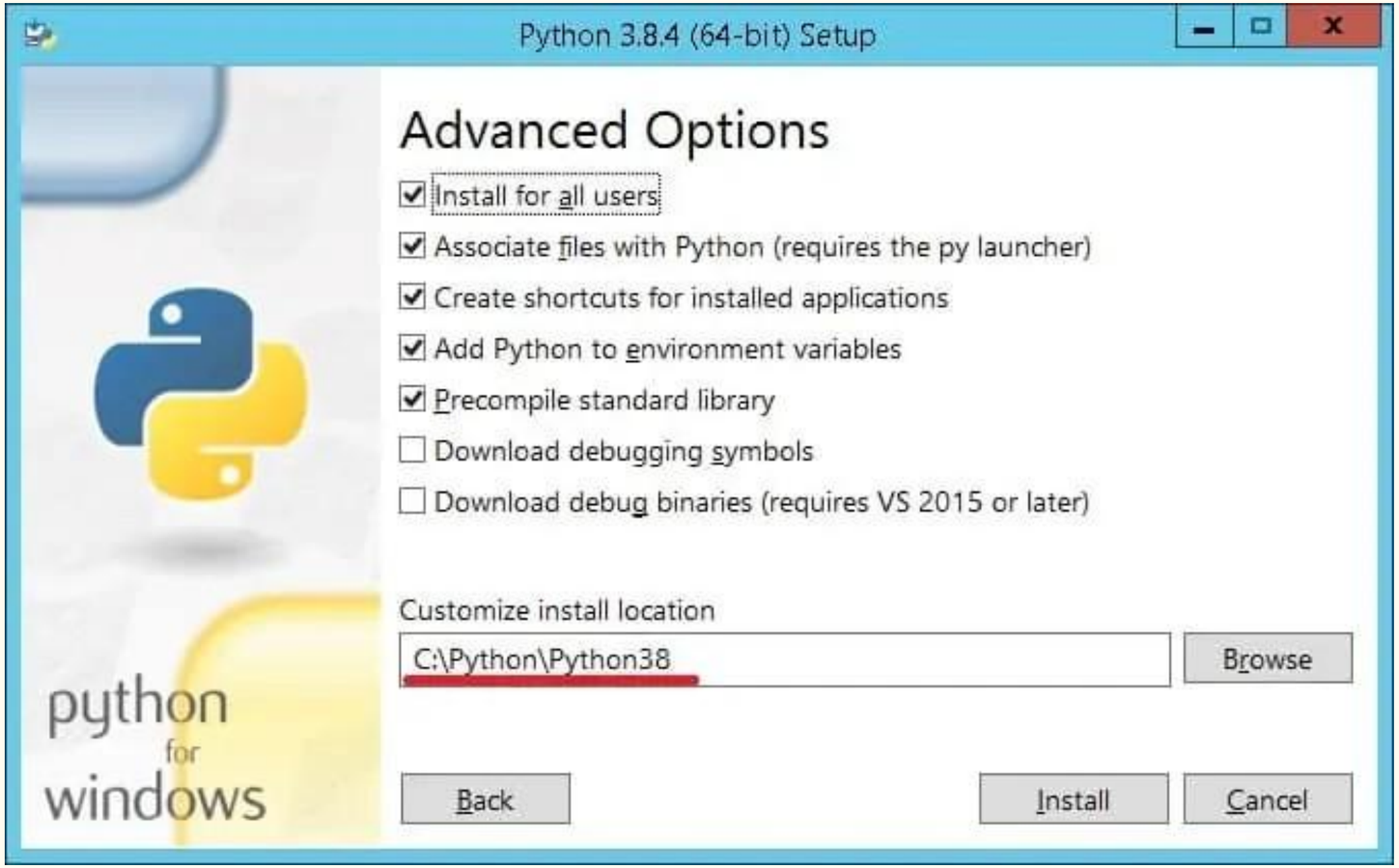
Accedi al sito web di Python e scarica l'ultima versione del programma di installazione di Python.

-Come amministratore, avviare l'installazione di Python.

-Selezionare entrambe le caselle di controllo nella parte inferiore dello schermo.

Fare clic sul pulsante per personalizzare l'installazione di Python.





-Selezionare tutte le caselle di controllo e fare clic sul pulsante Avanti.

-Selezionare la casella di controllo denominata: Installa per tutti gli utenti.

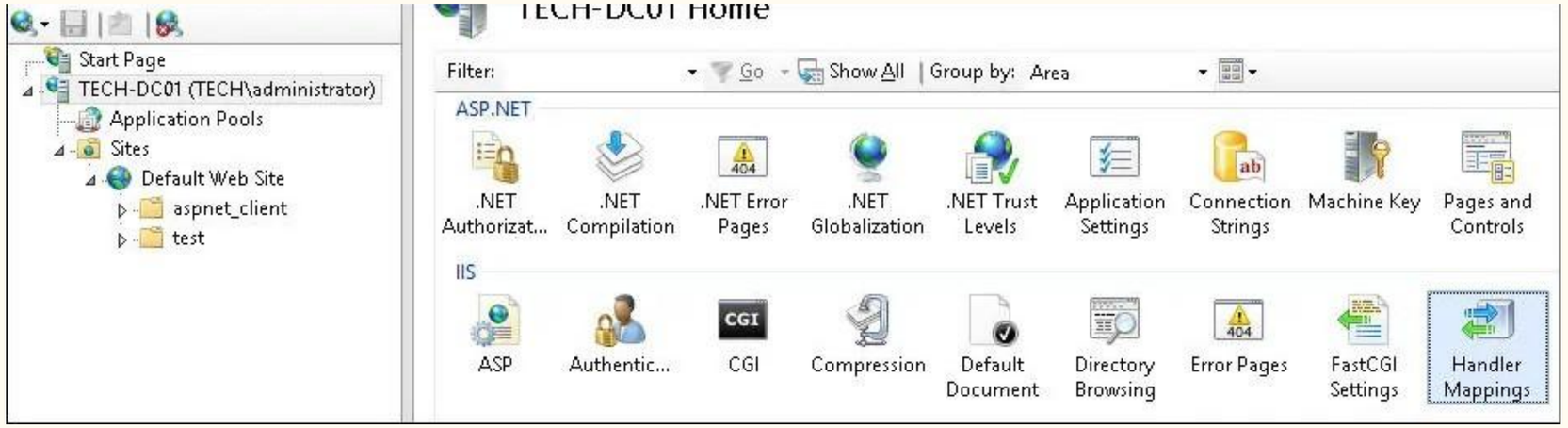
-Modificare il percorso di installazione di Python nella radice dell'unità C.

Esercitazione IIS - Abilitare Python sul server IIS:

-Avviare l'applicazione denominata: Gestione IIS.

-Nell'applicazione Gestione IIS, selezionare il nome del server IIS.

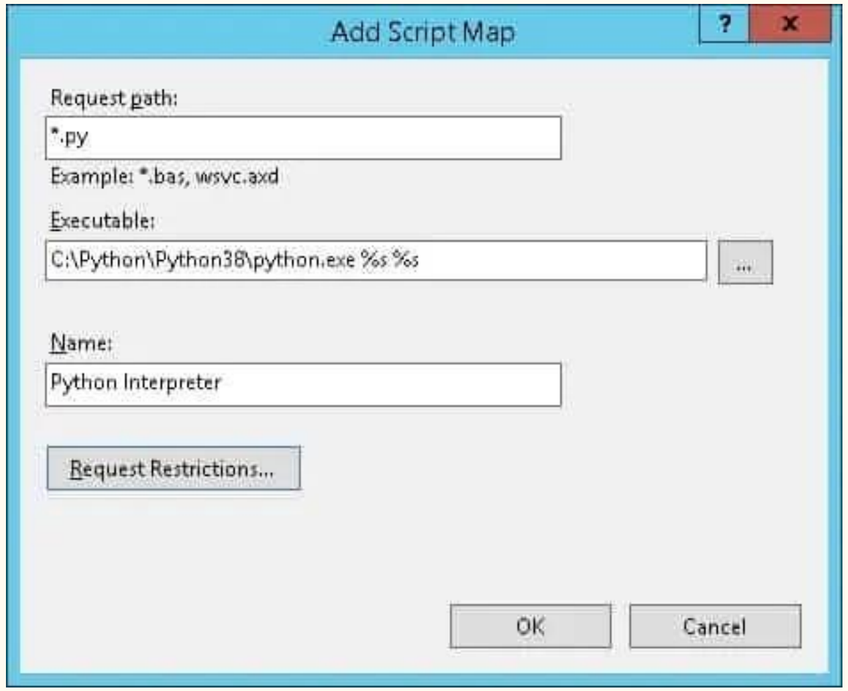
Nella parte destra dello schermo, accedere all'opzione denominata: Mapping gestori



Selezionare l'opzione denominata: Aggiungi mappa script

Eseguire la seguente configurazione:

- Request Path - *.py
- Executable - C:\Python\Python312\python.exe %s %s
- Interpreter - Python Interpreter Fare clic sul pulsante OK.



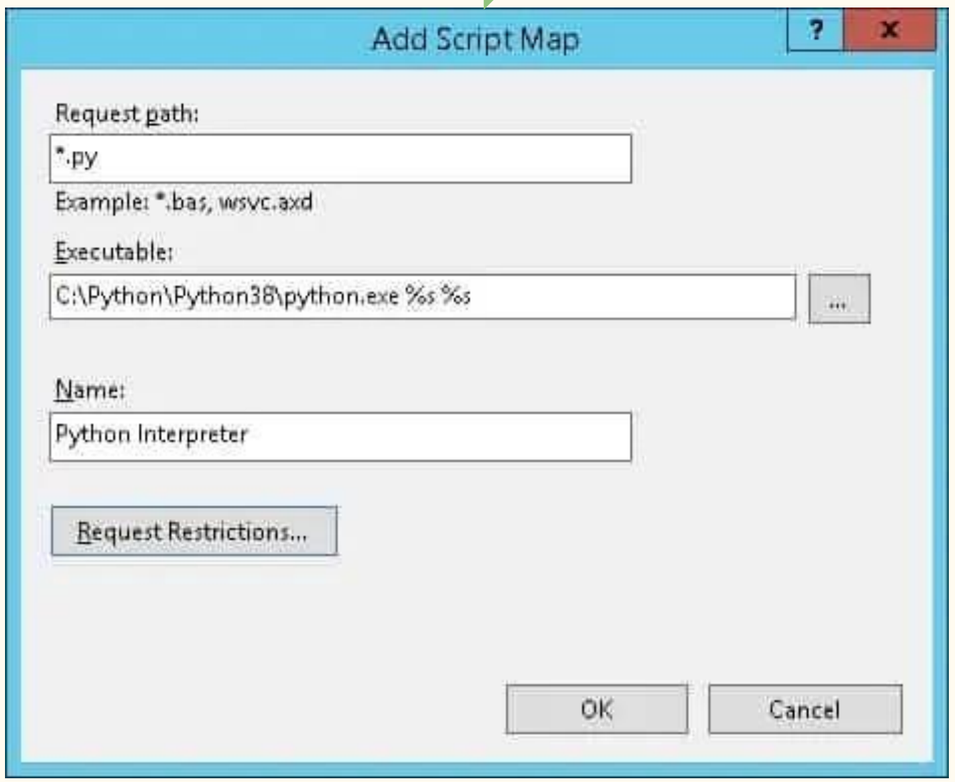
-Se viene visualizzato il seguente messaggio, fare clic sul pulsante Sì.

-Riavviare il servizio IIS.

-Python è stato abilitato correttamente sul server IIS.

Eseguire la seguente configurazione:

- Request Path - *.py
- Executable - C:\Python\Python312\python.exe %s %s
- Interpreter - Python Interpreter Fare clic sul pulsante OK.



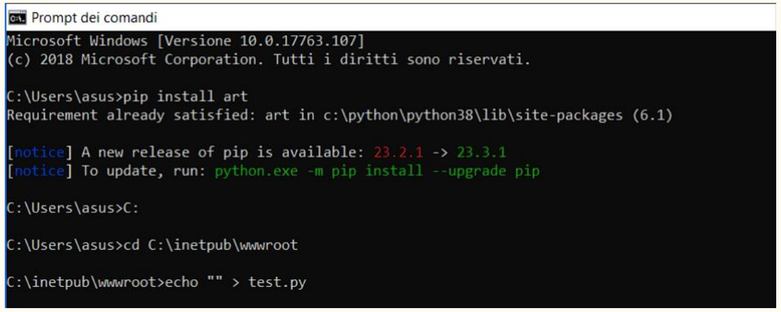
Se viene visualizzato il seguente messaggio, fare clic sul pulsante Sì.

Riavviare il servizio IIS. Python è stato abilitato correttamente sul server IIS.

Verifica Python

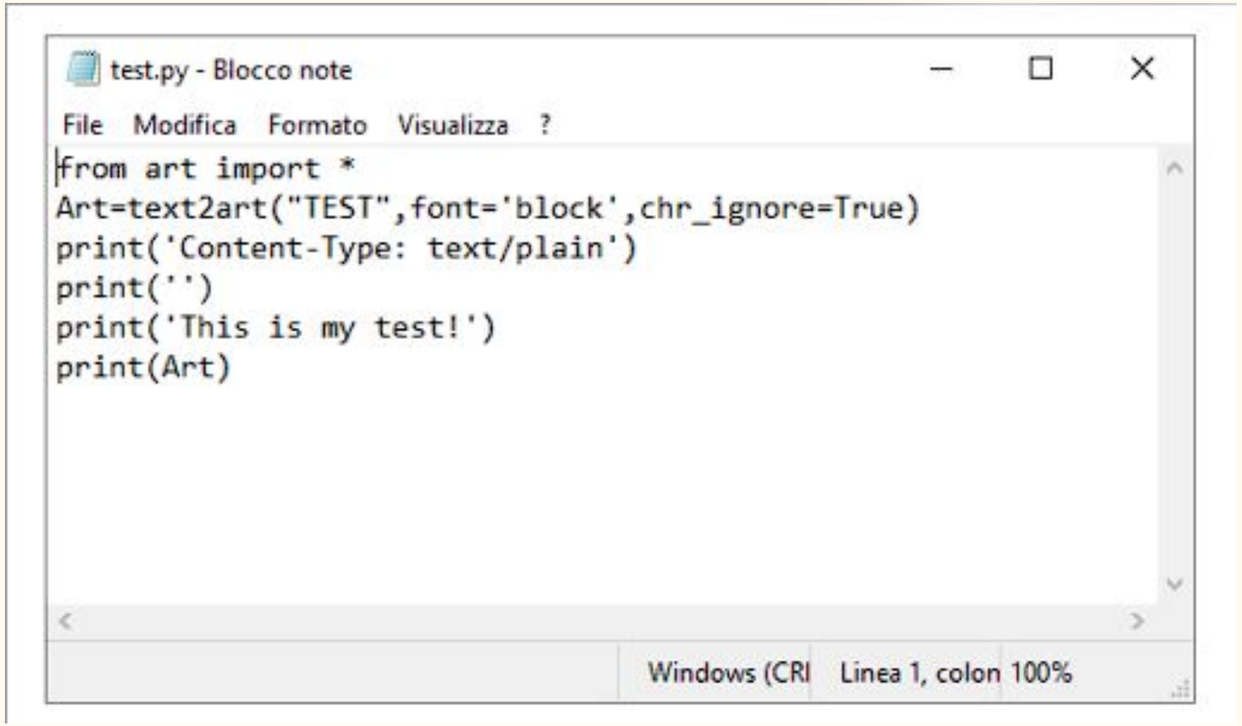
1. Eseguire cmd come amministratore
2. Eseguire il comando:

pip install art



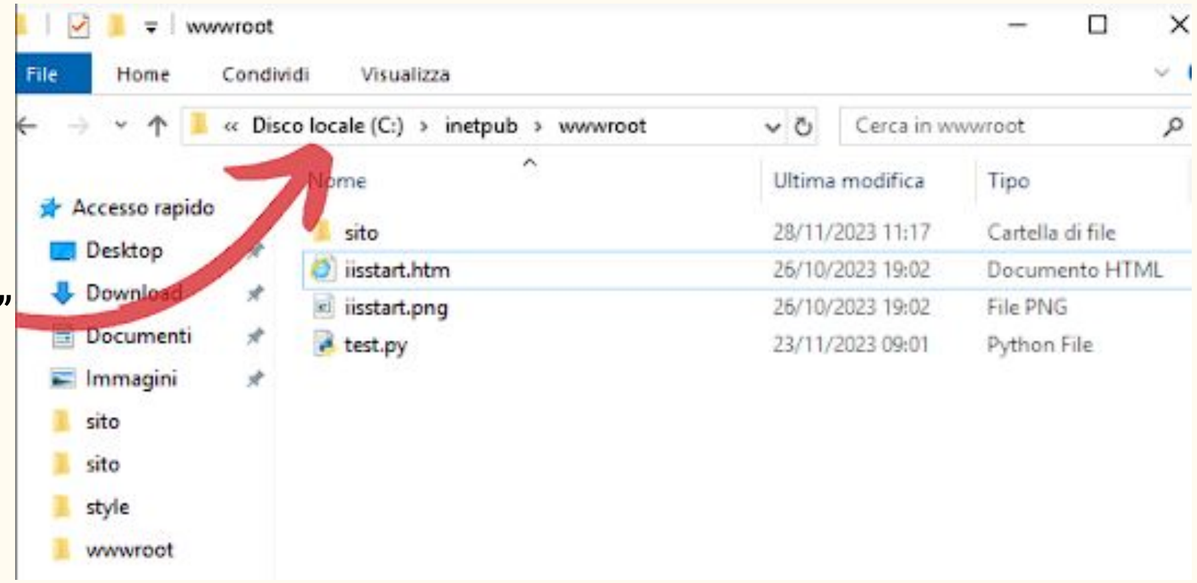
3. Creare un file.py con blocco note e mettere all'interno:

```
from art import *
Art=text2art("TEST",font='block',chr_ignore=True)
print('Content-Type: text/plain')
print('')
print('This is my test!')
print(Art)
```

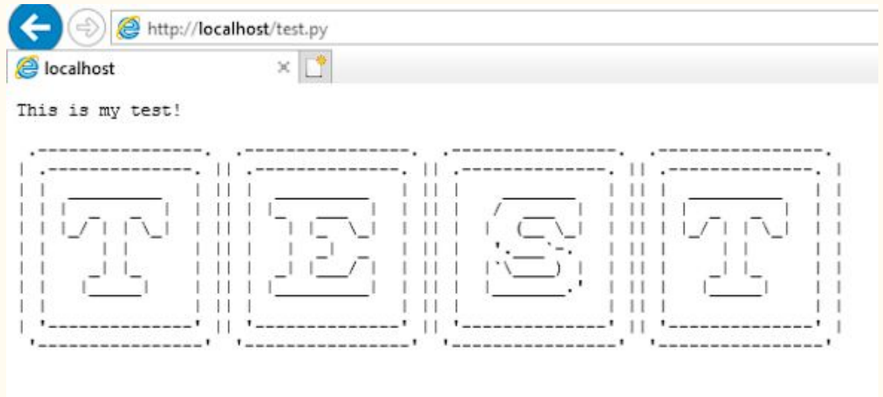


Verifica Python

Spostare il file nella cartella “wwwroot”



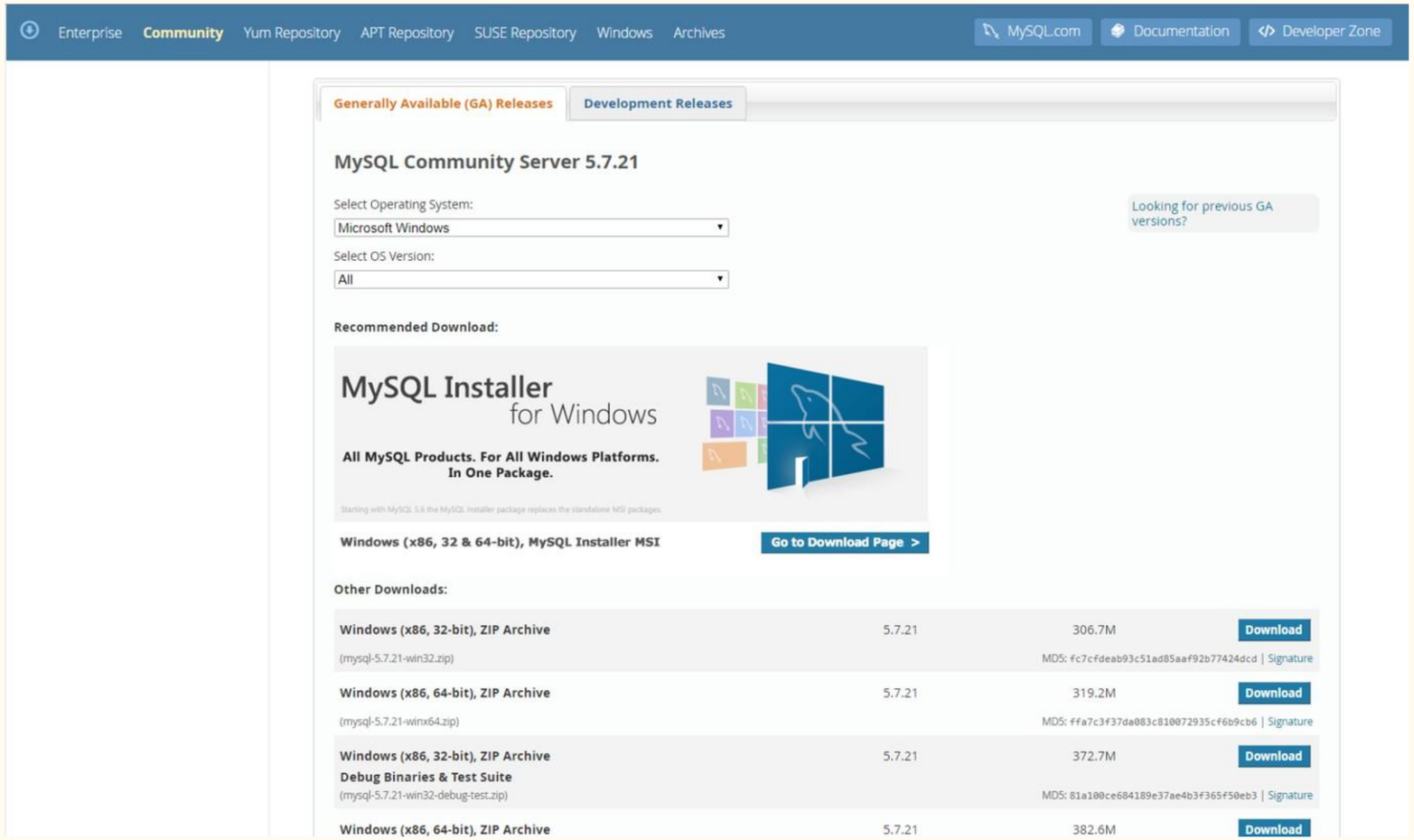
Per eseguirlo: immettere l’indirizzo indicato, nella barra di ricerca del browser



Come installare MySQL in Windows 10

MySQL™

Download del software necessario:



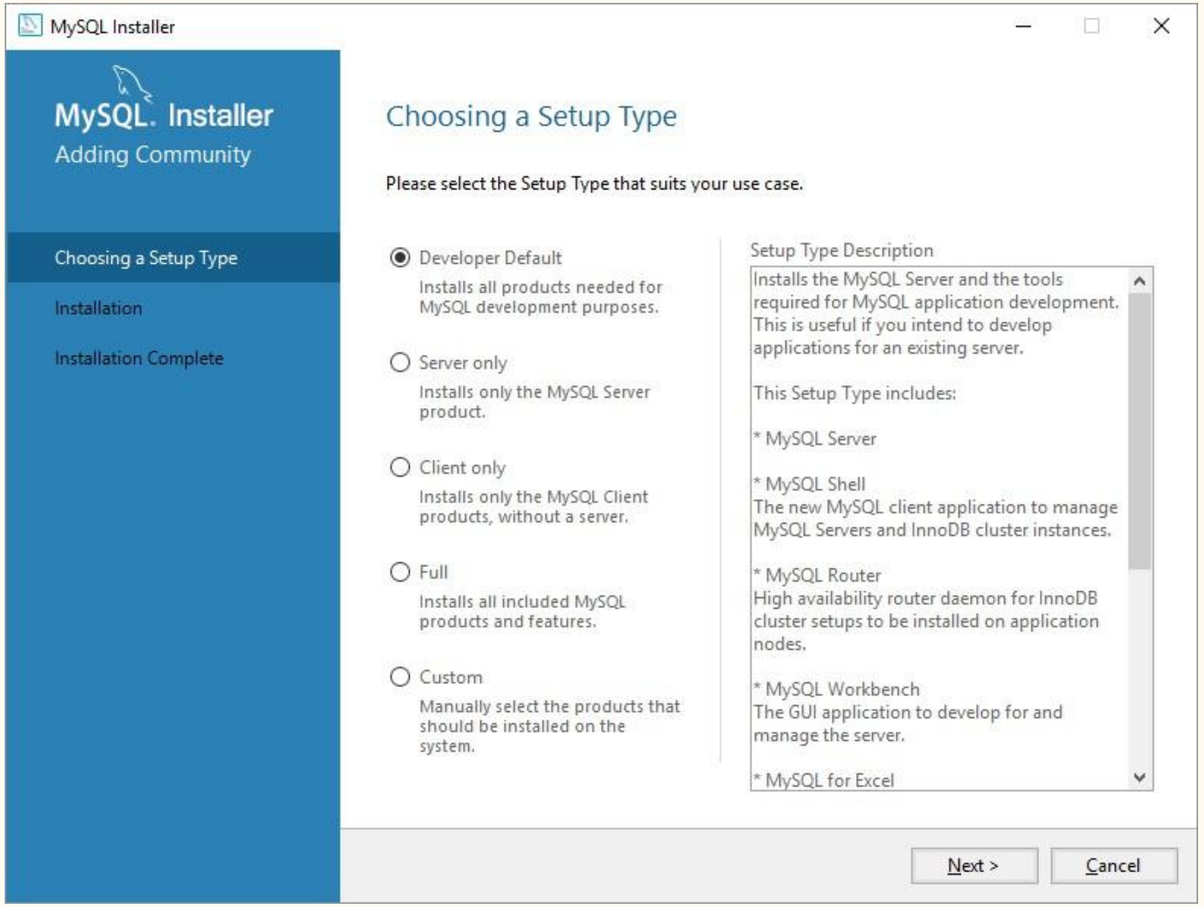
-Per scaricare il pacchetto di installazione, puntare la pagina del browser all'indirizzo: https://dev.mysql.com/ downloads/mysql/.

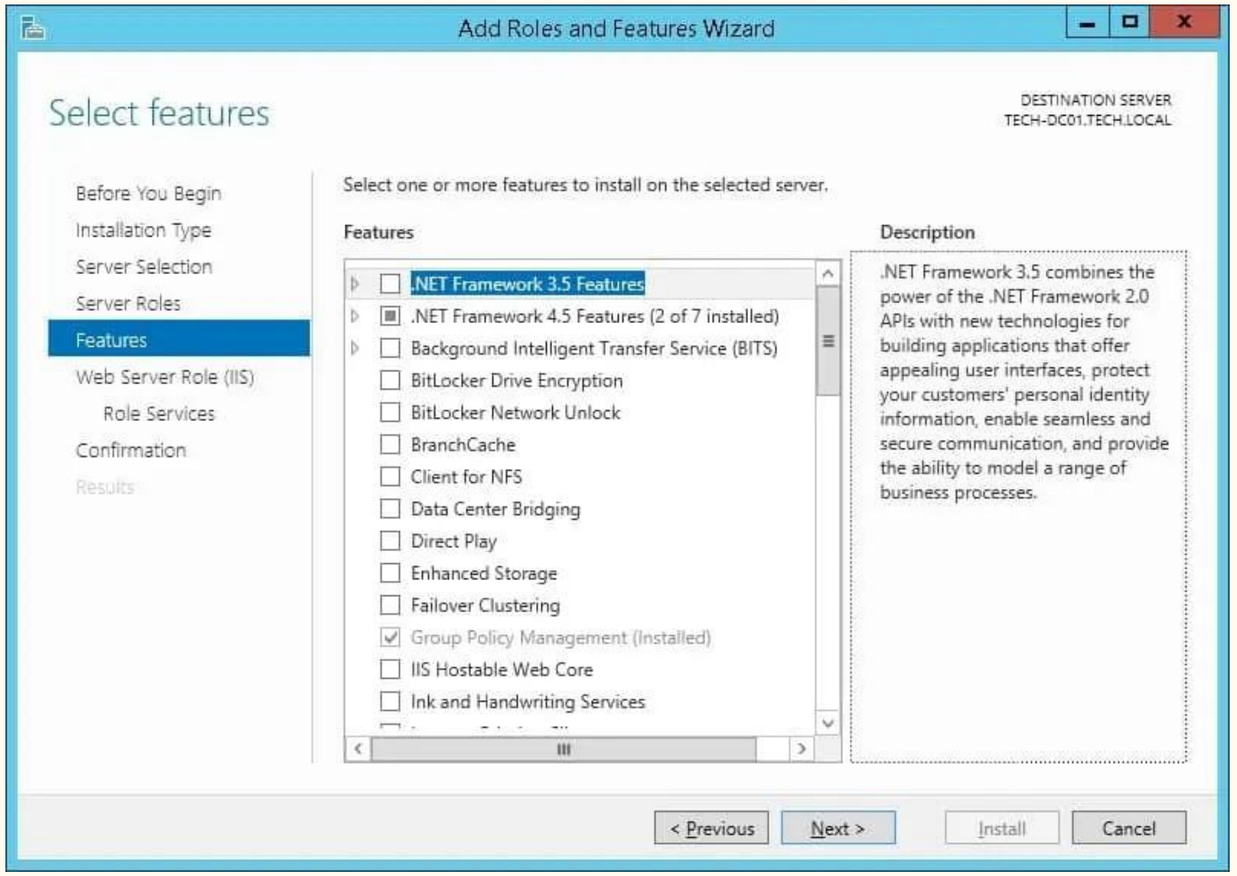
Installazione:

-Terminato il download, potrai finalmente avviare l’installazione del pacchetto eseguendo il file EXE appena scaricato.

-Appena avviato, ti verrà presentata questo pannello, tramite il quale potrai verificare il tipo di installazione desiderata.

-Installare la versione “Full”





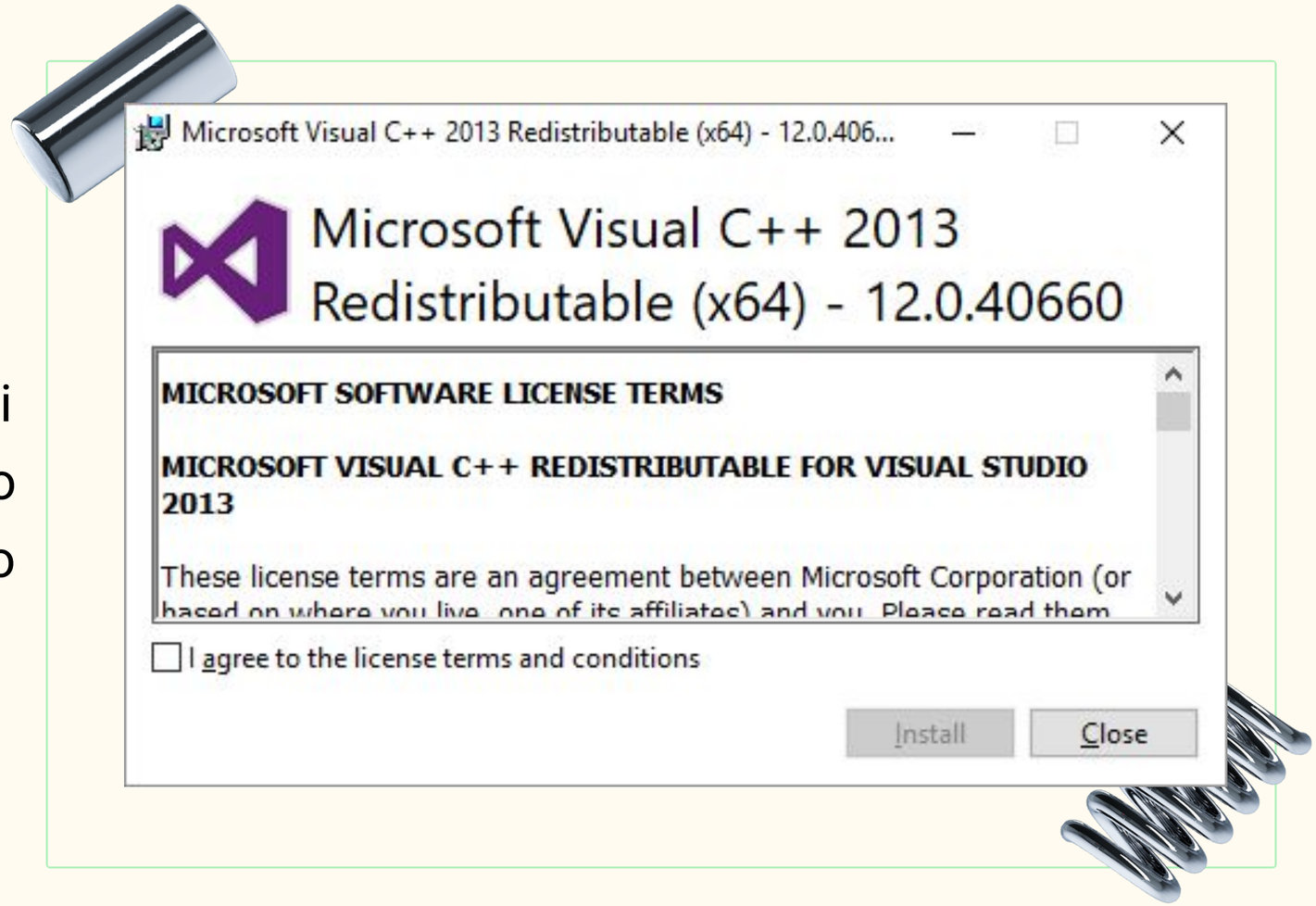
La prossima schermata che ti verrà mostrata, serve all'installatore (il pacchetto exe che hai appena avviato) per verificare che il tuo sistema abbia tutto ciò che gli serve per fare un'installazione coerente.

-Pertanto qui potrai cliccare su "Execute" per fare in modo che vengano risolte tutte le dipendenze per l'installazione.

-Durante questa fase, che può prendere alcuni minuti, verranno scaricati tutti i pacchetti necessari per risolvere le dipendenze.

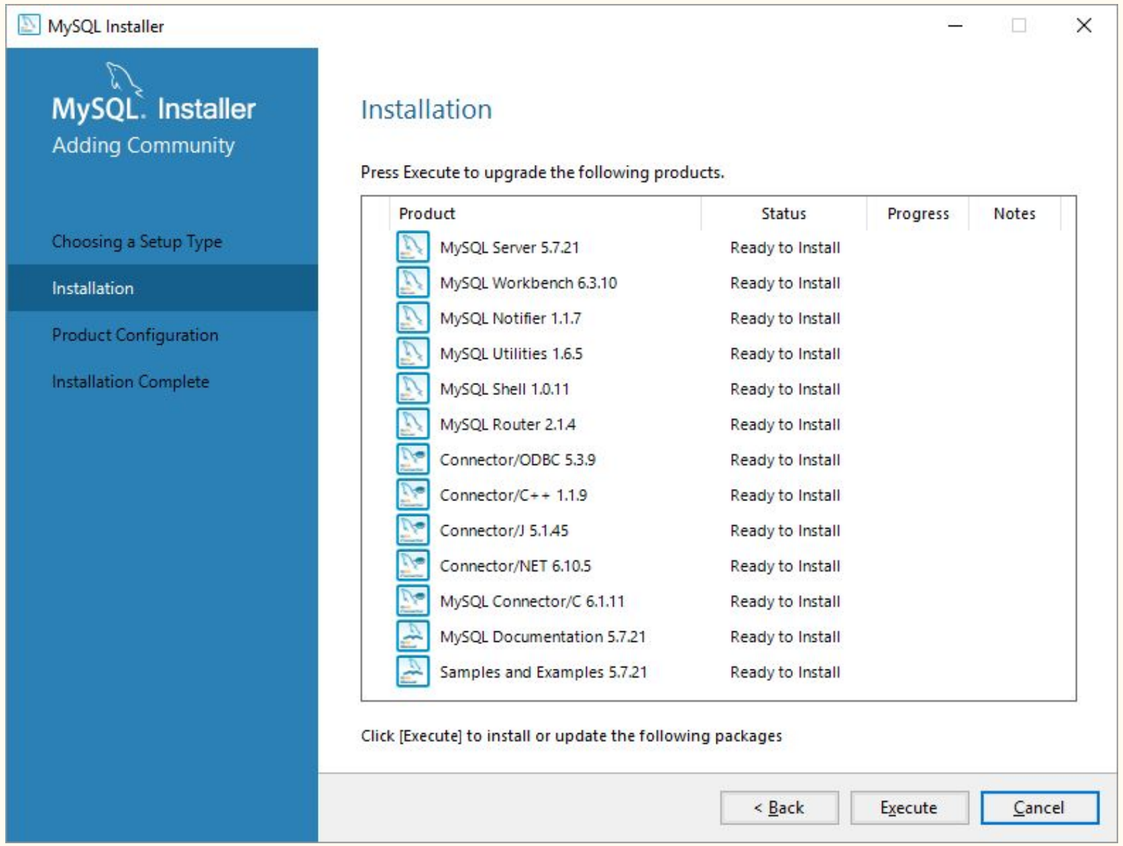
-Una delle richieste di conferma per l’installazione automatica delle dipendenze.

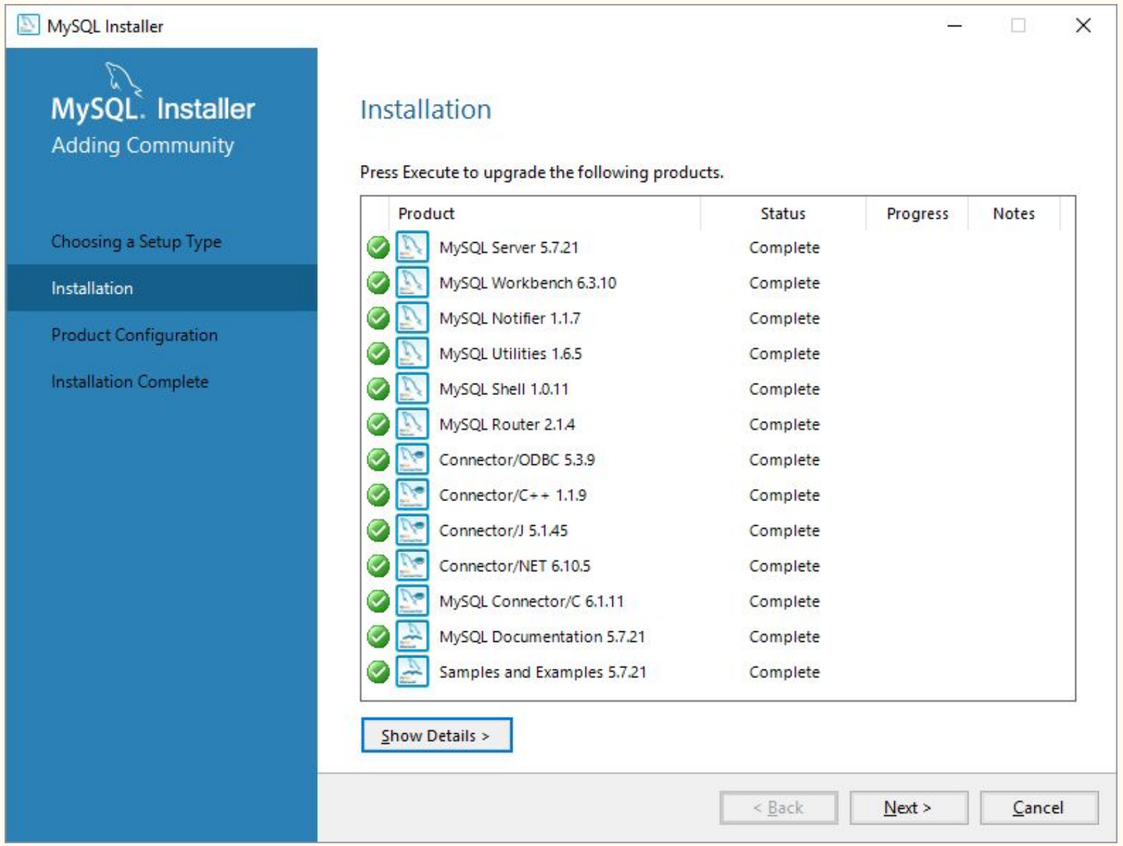
-Tuttavia, potrebbero essere necessari diversi passi simili a questo, che dipendono fondamentalmente da cosa viene rilevato mancante nel tuo sistema Windows.



-Per avviare, premere Execute.
-Quando questa fase sarà terminata ti verrà mostrato un pannello simile al precedente, che ti indicherà se qualcosa è andato storto oppure no.

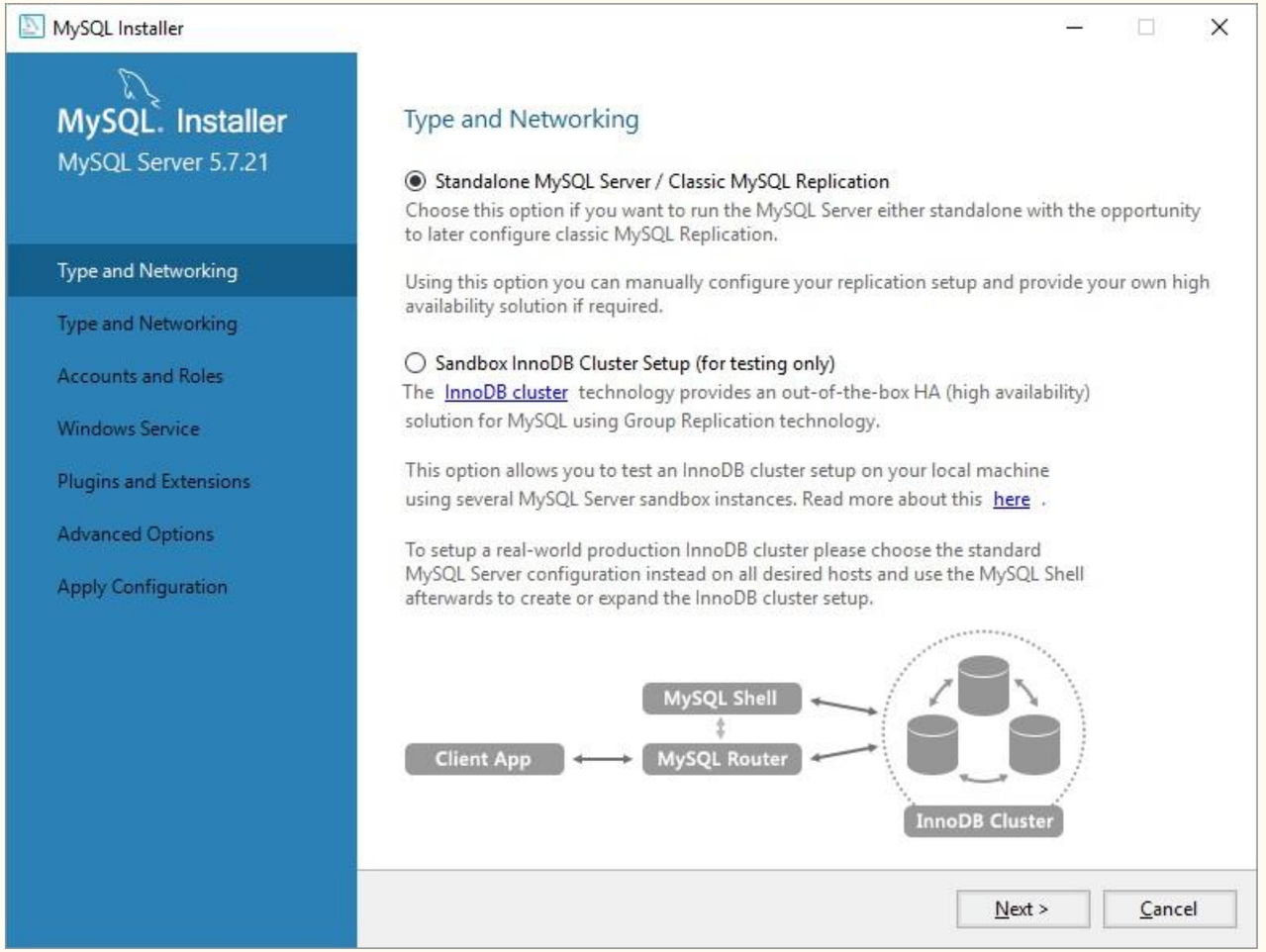
-Premi Next per completare questa fase dell'installazione.





Configurazione:

Una volta completata questa fase, l’installatore non terminerà, ma provvederà a guidarti nella fase successiva per la completa configurazione dei pacchetti appena installati, mostrandoti questa schermata iniziale.
-Premendo “Next” potrai proseguire con tutte le configurazioni necessarie.



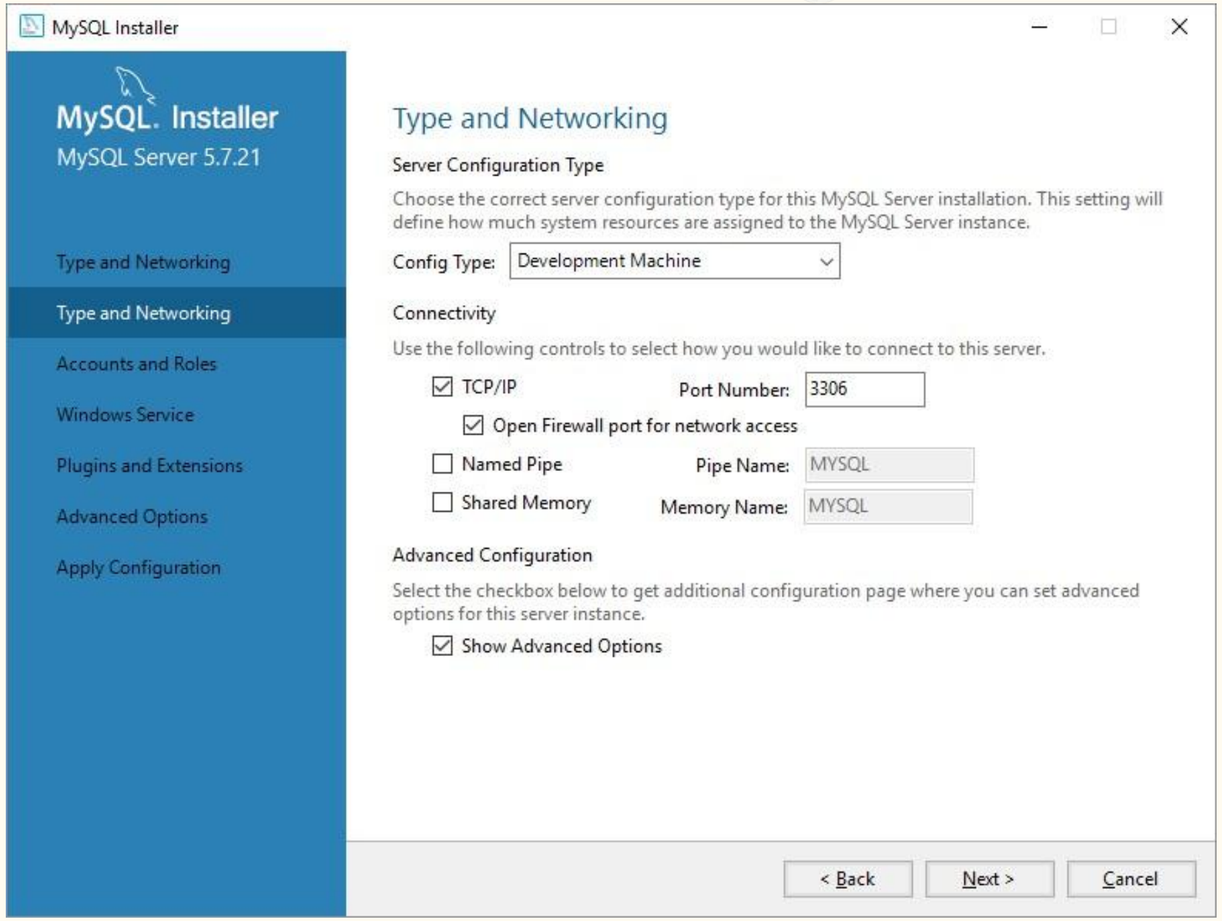
Tipo di Server e Rete:

.Qui potrai configurare sia il tipo di server che soprattutto la rete.

-Tipo di configurazione, scelta per il nostro Mysql:

- Development Machine – Sistema di sviluppo con molte applicazioni accessorie.

-Nello stesso pannello di configurazione, dovrai specificare il tipo di connessione che vuoi rendere disponibile per poter accedere a MySQL, noi non abbiamo modificato nulla



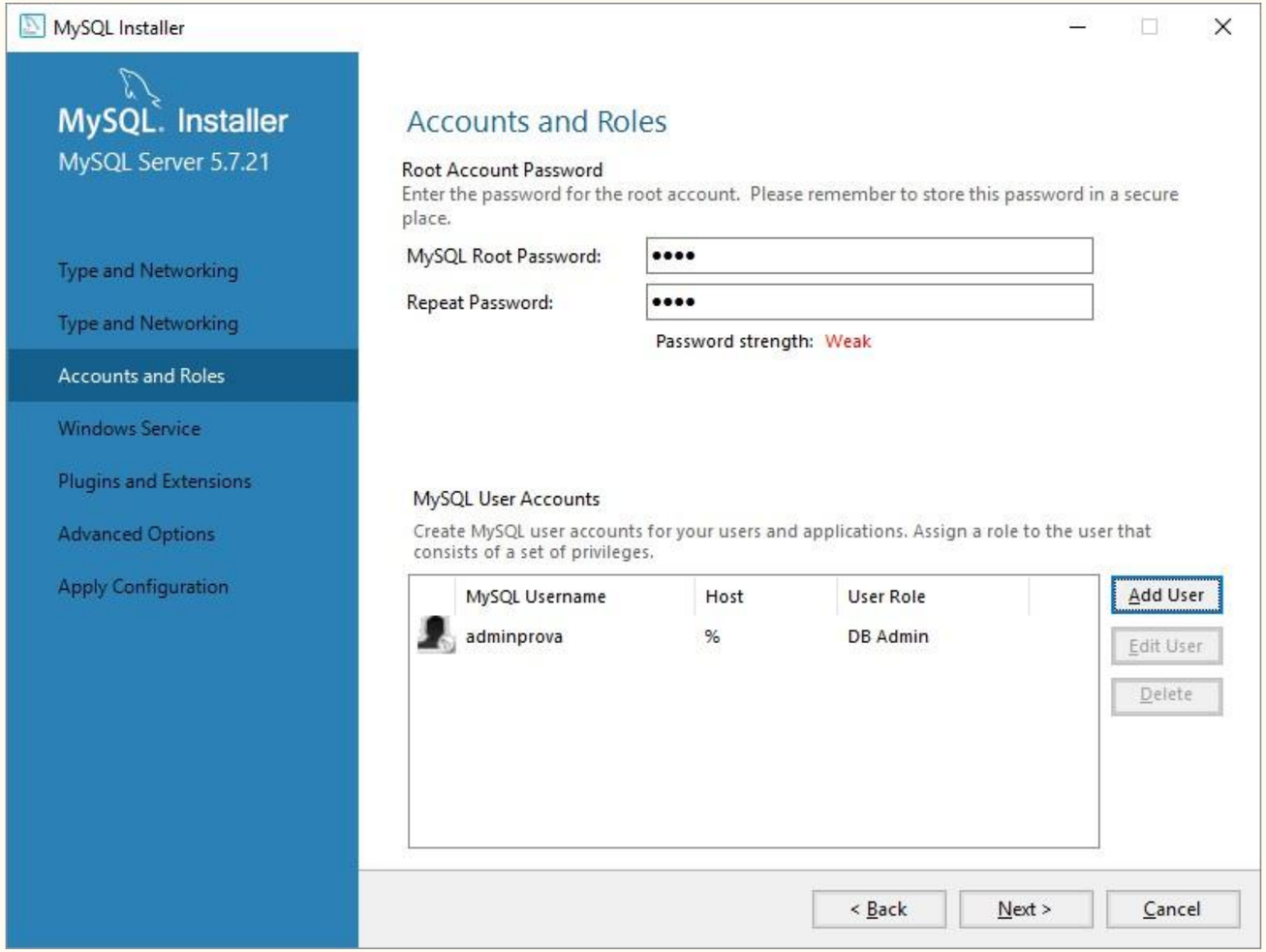
Account e Ruoli:

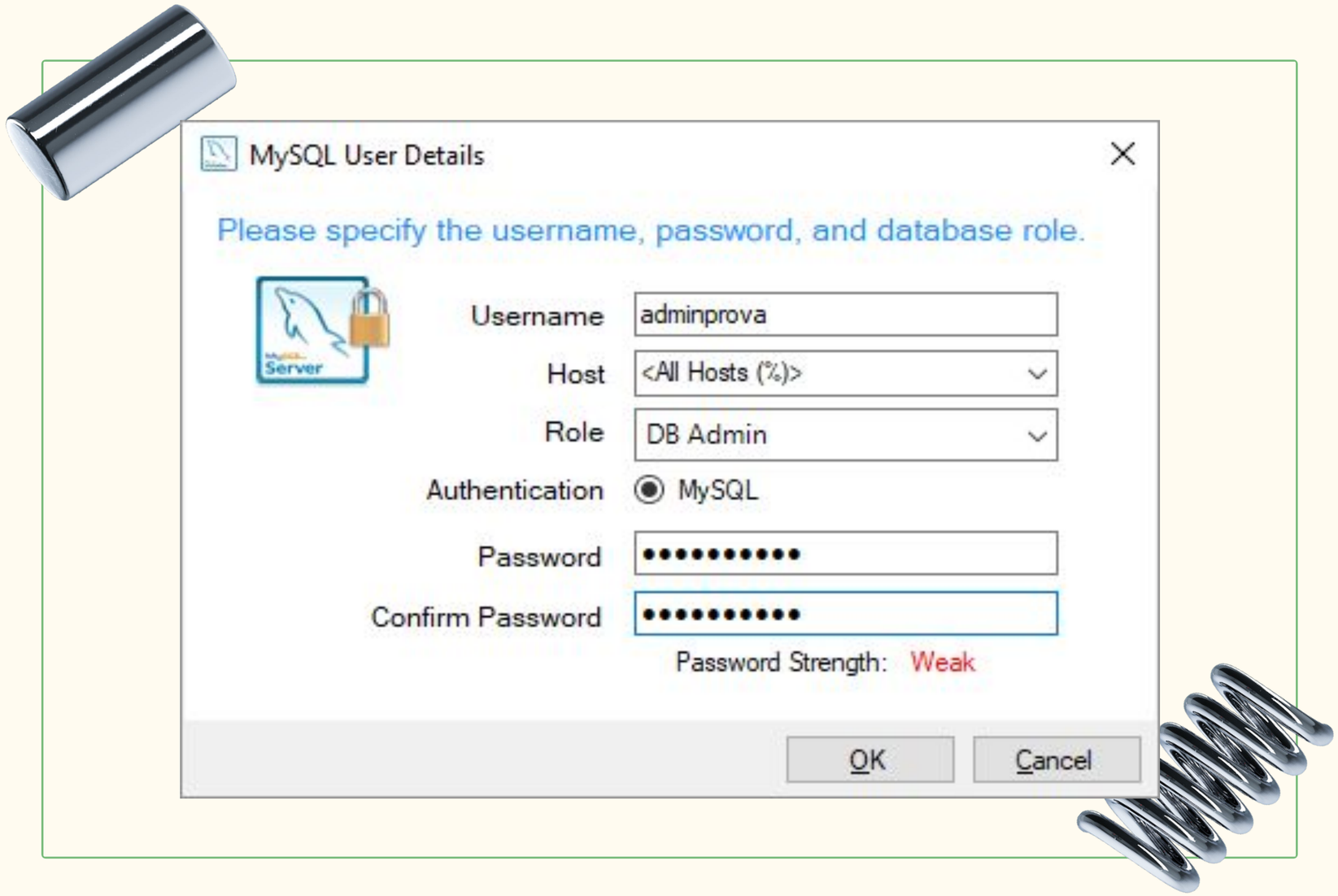
-Per poter accedere a MySQL bisogna assolutamente creare account e ruoli specifici.

-In questo passo, puoi definire gli account ed i ruoli per l’accesso a MySQL.

-La prima cosa da fare in assoluto è defiire la Root password, nella parte alta del pannello.

-Mentre potrai sfruttare la parte inferiore del pannello per creare tutti gli account che desideri, cliccando su Add User e specificando le credenziali per il nuovo account che stai creando.

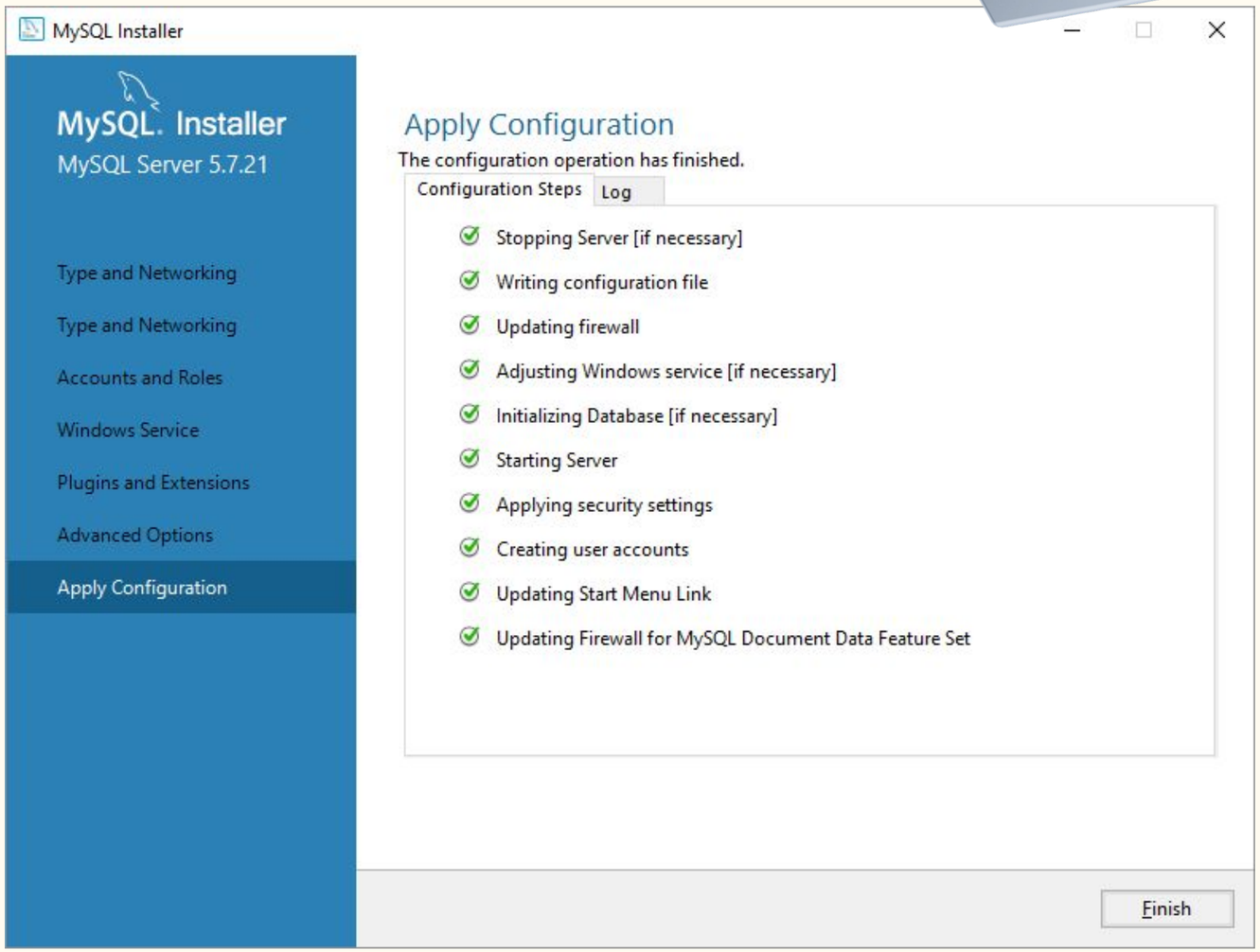




Premi Next per accedere al passo successivo.

Apply Configuration:

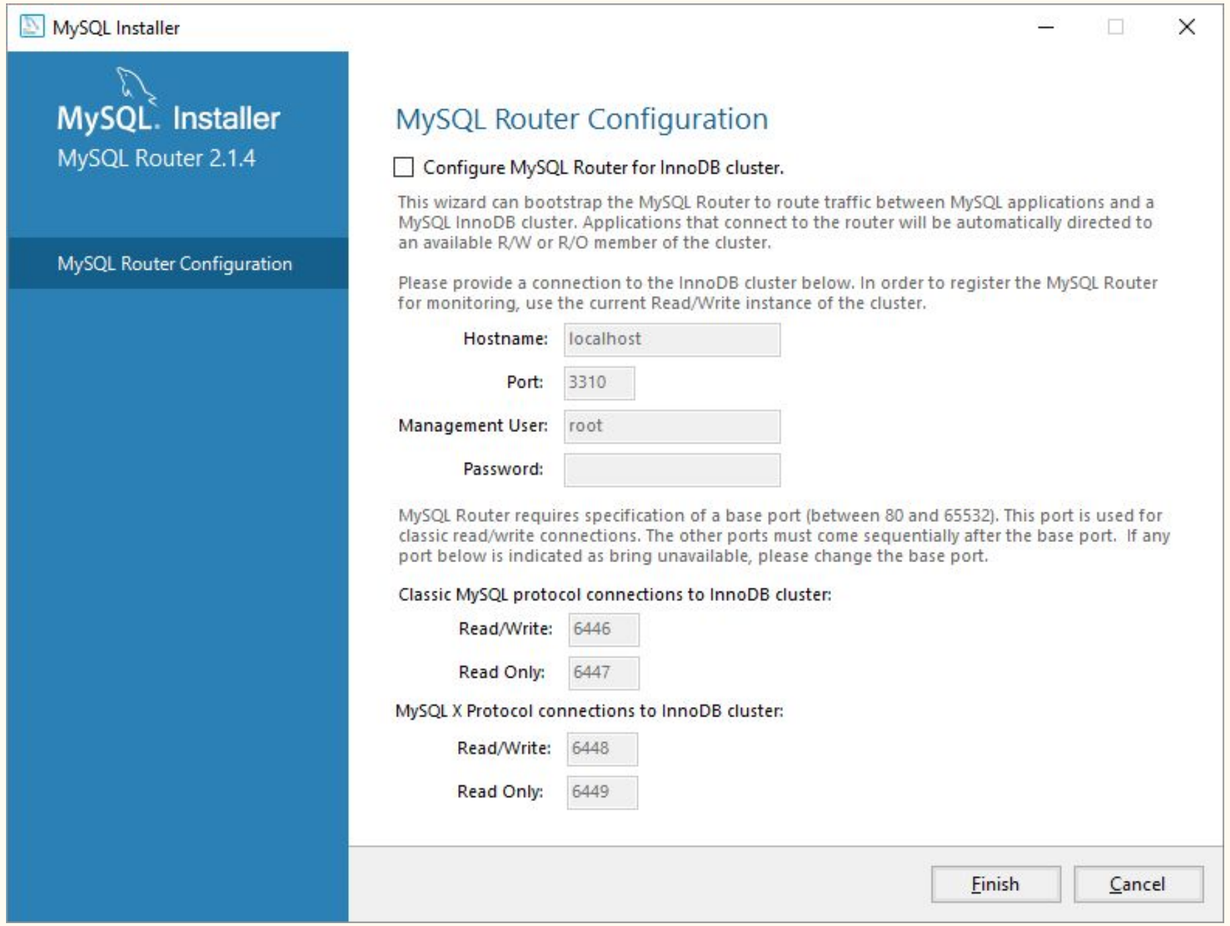
-L’ultimo passaggio della configurazione, consiste nell’applicare tutto ciò che hai specificato fin qui.
-Assicurati che dopo aver premuto Execute, il risultato finale non presenti errori e sia simile all’immagine qui sopra.
-Se è tutto OK, direi che puoi passare al passo successivo, premendo Finish.



MySQL Router Configuration:

-In questo pannello ti verrà chiesto se vuoi attivare MySQL Router:

-Se non sai di che cosa sto parlando, non importa. Puoi semplicemente lasciare MySQL Router disabilitato e premere Finish.

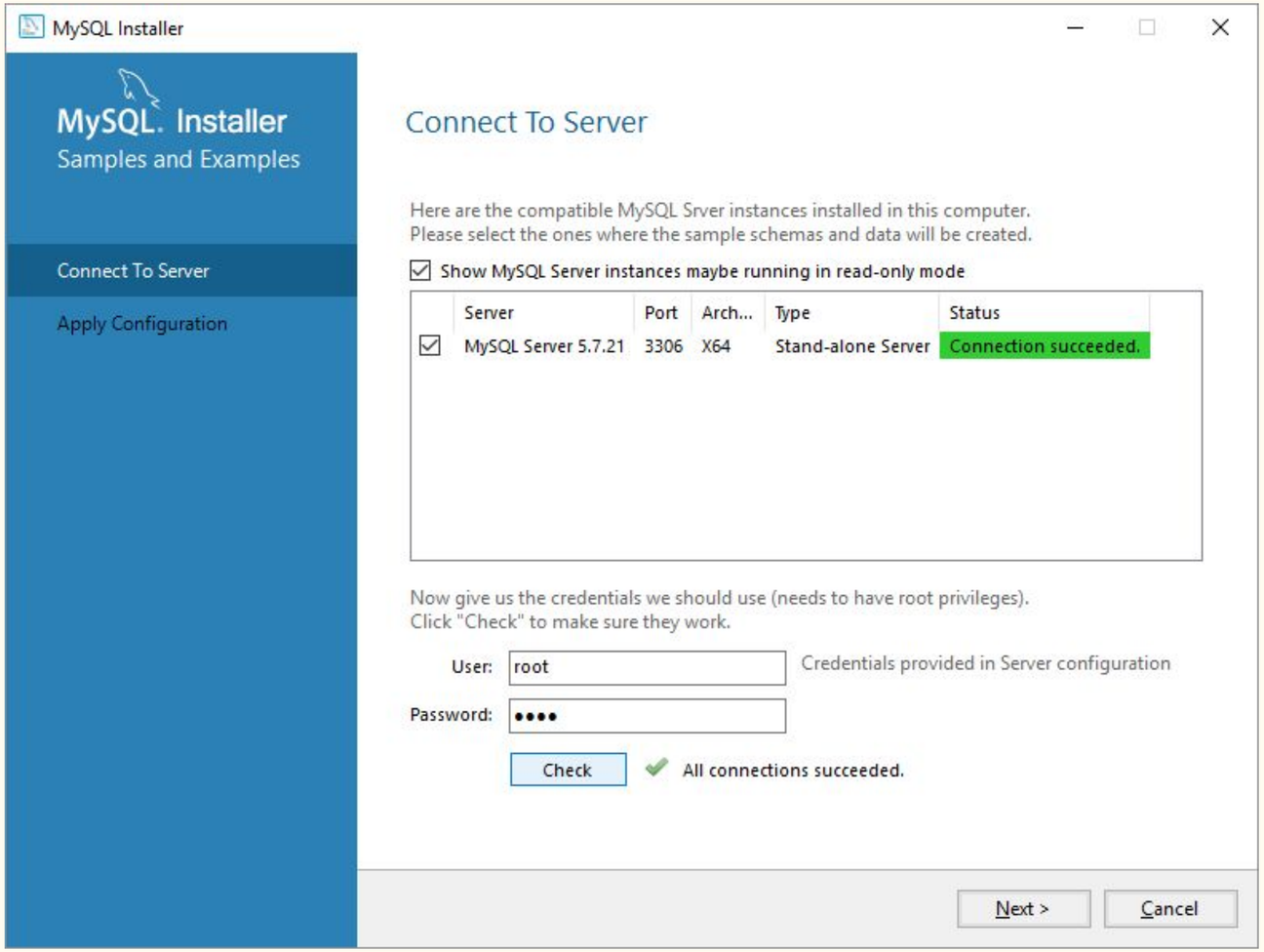


Connect to Server:

-Ora che tutto è installato, sarebbe utile verificare se la connessione al server funziona correttamente.

-Inserendo le credenziali degli account precedentemente definiti, puoi verificare la tua connessione al database.

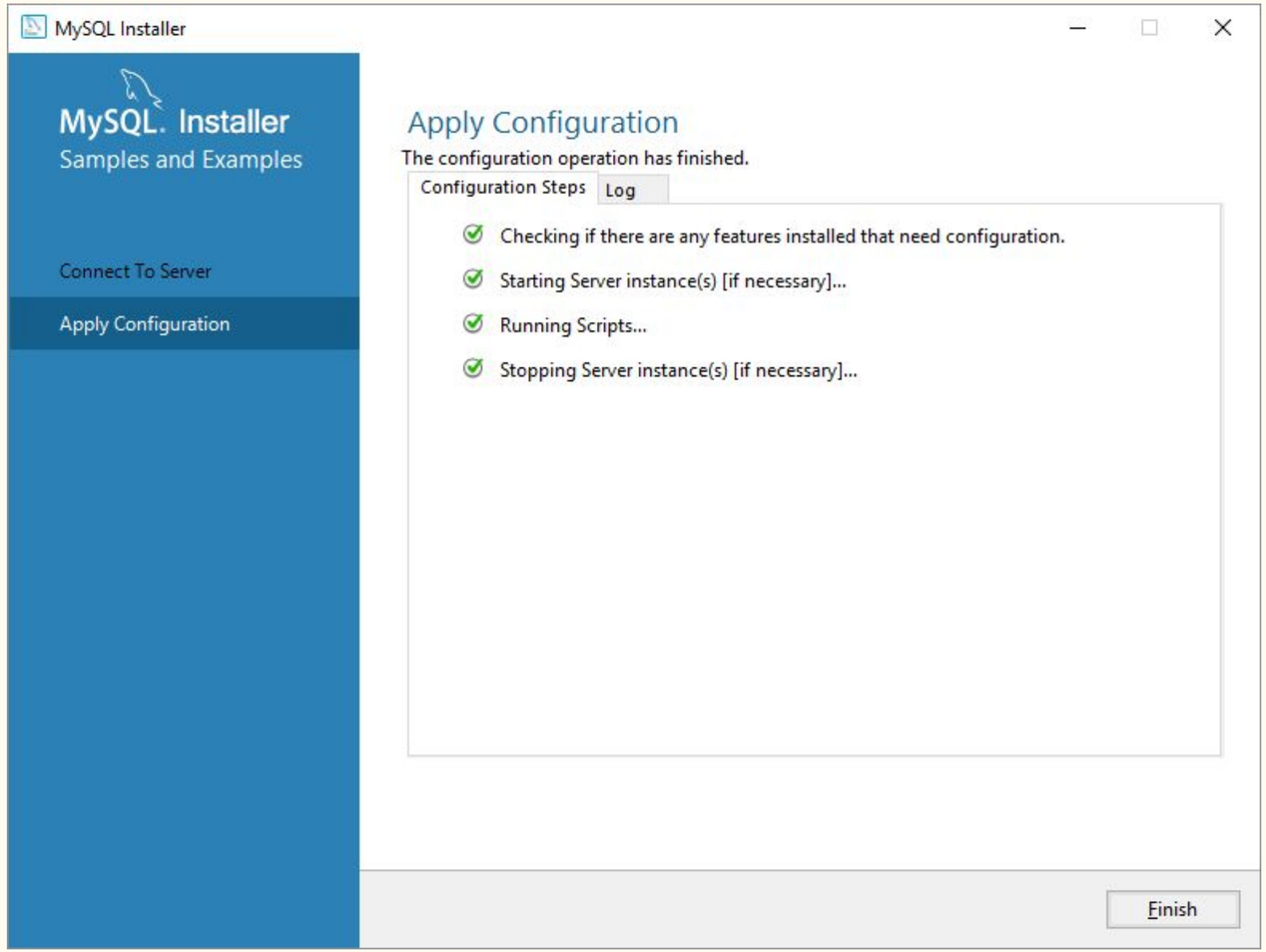
-Se il test di connessione non ha problemi ti verrà segnalato con: Connection succeded , ad indicare che tutto è andato bene.



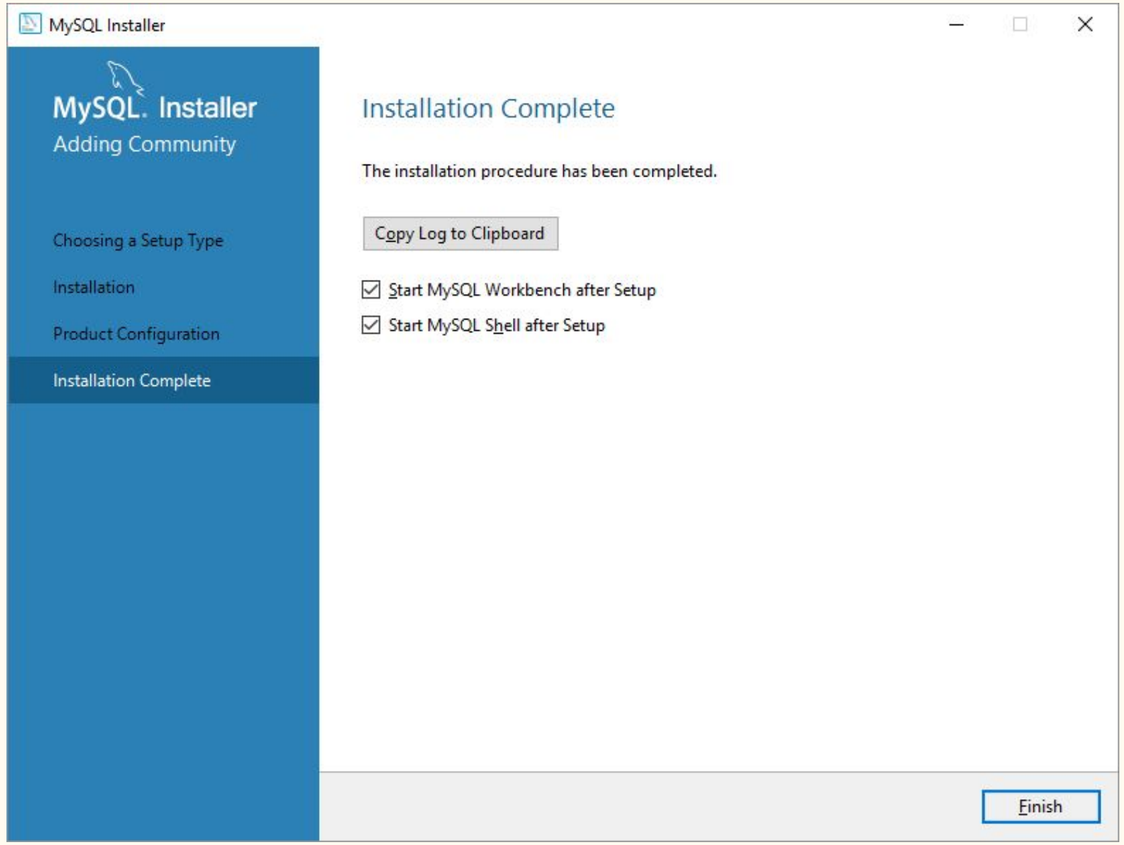
Apply Configuration (di nuovo):

-In modo che tutti i servizi vengano riavviati con le nuove definizioni.

-Premi il tasto Finish per passare all’ultima schermata.



Installation Complete:



Premi Finish per chiudere l’installatore ed avviare un terminale MySQL e Mysql Workbench, lo strumento di sviluppo per MySQL.

PROGETTO:

Login:

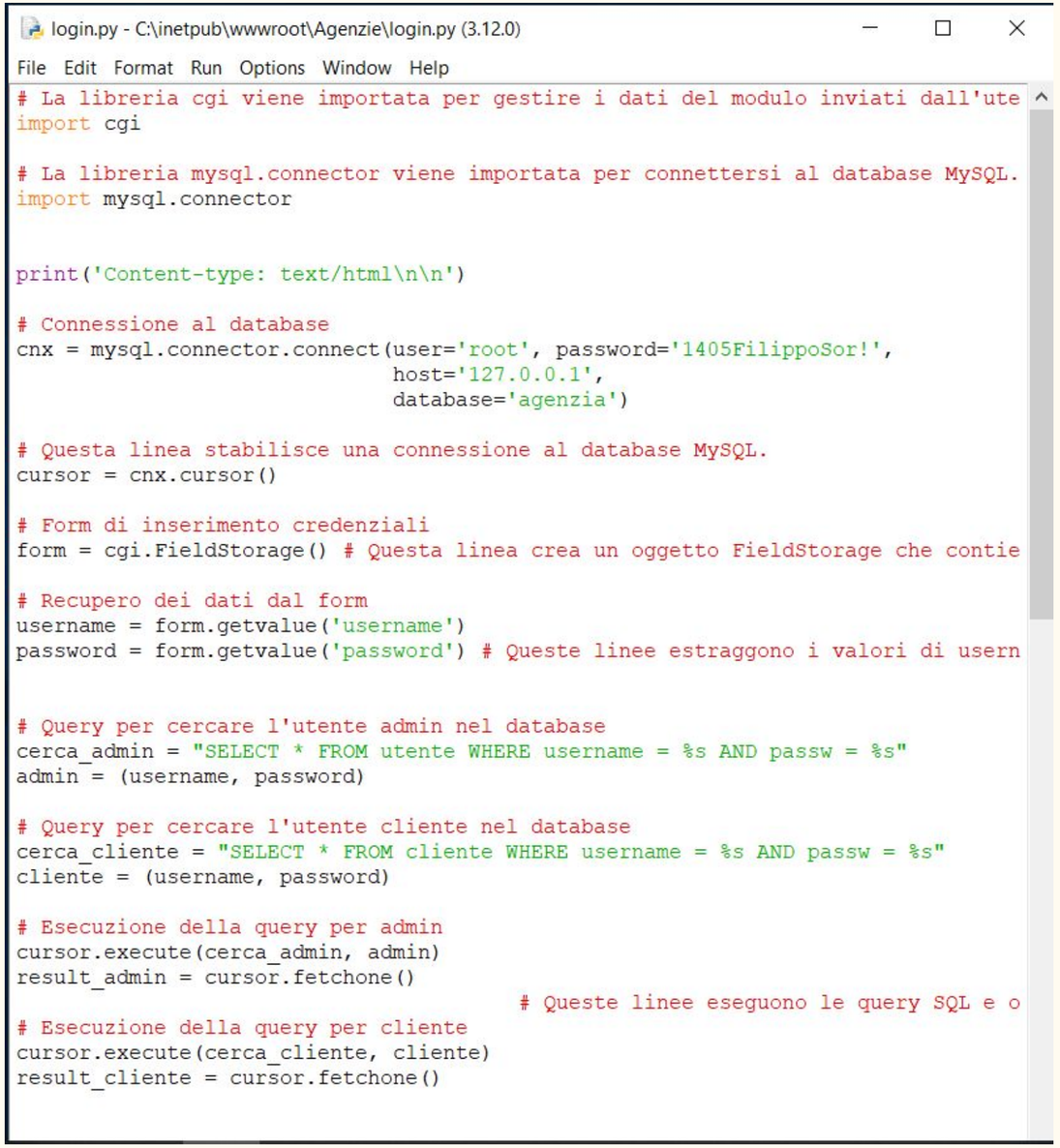
Questo codice HTML è un modulo di login. Quando un utente visita la pagina, vedrà due campi di input, uno per il nome utente e uno per la password. Entrambi i campi devono essere compilati per poter inviare il modulo, grazie all'attributo "required". Una volta che l'utente ha inserito le sue credenziali e ha cliccato sul pulsante di invio, i dati vengono inviati al server per essere elaborati da un file chiamato "login.py". Questo è fatto attraverso una richiesta HTTP POST, che invia i dati come corpo della richiesta, piuttosto che come parte dell'URL. Inoltre, il codice include un collegamento a un foglio di stile esterno ("style.css") per la formattazione e la presentazione della pagina.

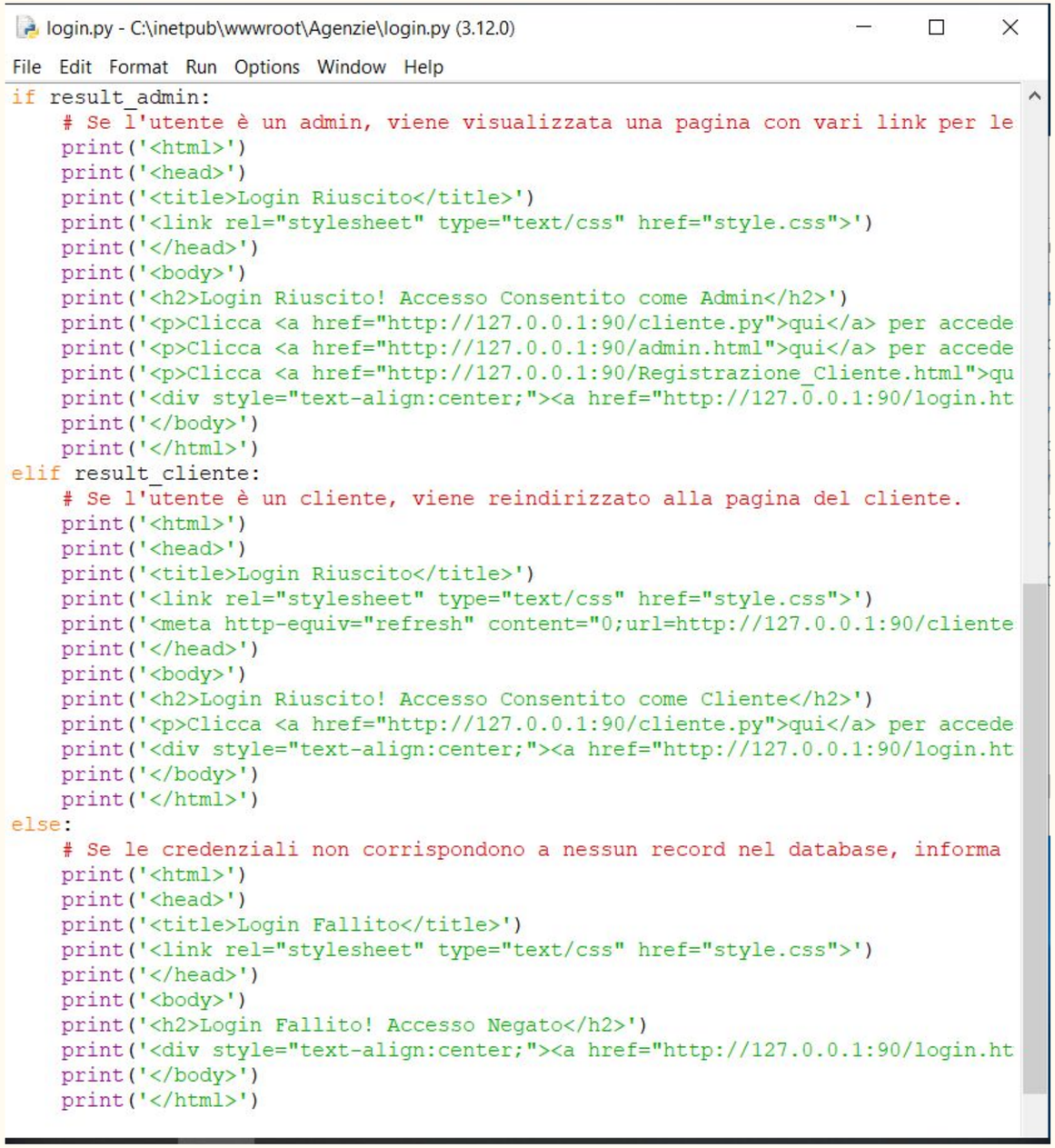
login.html - Blocco note

File Modifica Formato Visualizza ?

```
<!DOCTYPE html>
<html lang="it">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" type="text/css" href="style.css">
    <title>Login</title>
</head>
<body>
    <form action="login.py" method="post">
        <h2>Login</h2>
        <label for="username">Username:</label>
        <input type="text" id="username" name="username" required>
        <label for="password">Password:</label>
        <input type="password" id="password" name="password" required>
        <input type="submit" value="Login">
    </form>
</body>
</html>
```

Questo script Python è progettato per gestire il processo di autenticazione di un'applicazione web. Inizialmente, stabilisce una connessione con un database MySQL e recupera le credenziali di accesso inviate dall'utente attraverso un modulo web.
Successivamente, esegue due query SQL per cercare le credenziali dell'utente nelle tabelle 'utente' (per gli amministratori) e 'cliente' del database.





Se le credenziali corrispondono a un record nella tabella ‘utente’, l’utente viene riconosciuto come amministratore e viene indirizzato a una pagina con vari link per le funzionalità dell’amministratore. Se invece le credenziali corrispondono a un record nella tabella ‘cliente’, l’utente viene riconosciuto come cliente e viene reindirizzato alla pagina del cliente.

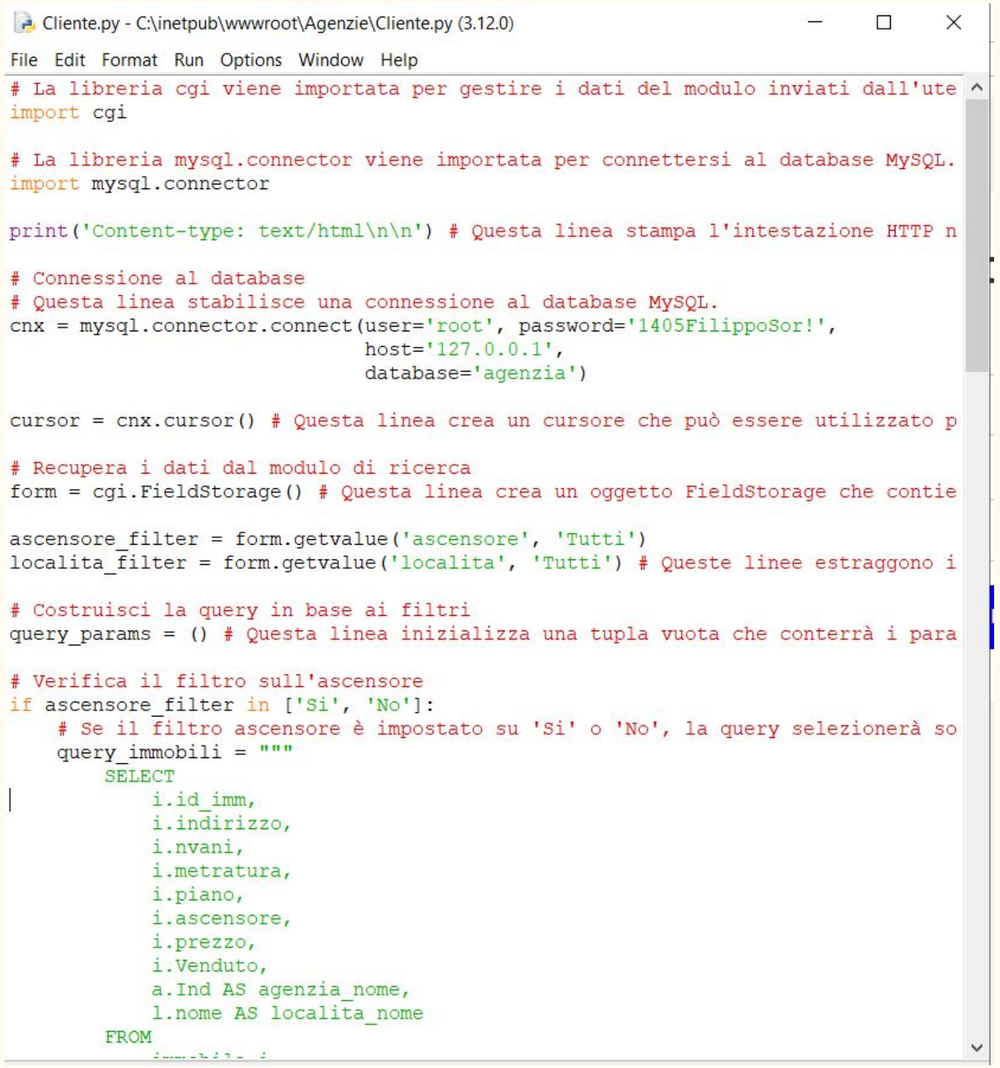
Nel caso in cui le credenziali non corrispondano a nessun record nel database, l’utente viene informato che l’accesso è fallito. Infine, il cursore e la connessione al database vengono chiusi per liberare le risorse.

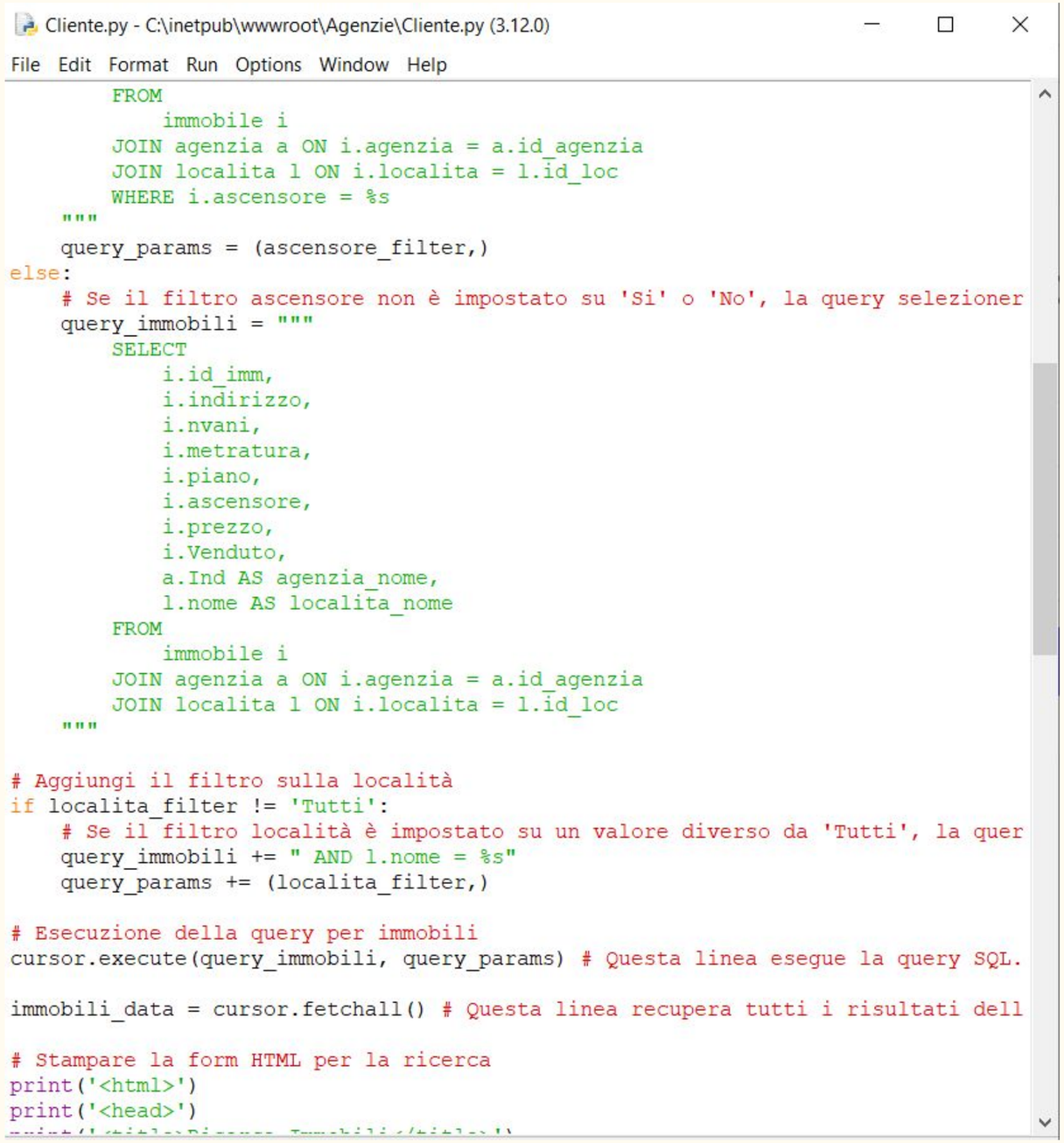
In sintesi, questo script gestisce l’autenticazione degli utenti, distinguendo tra amministratori e clienti, e reindirizza gli utenti alla pagina appropriata in base al loro ruolo. Se le credenziali fornite non sono valide, informa l’utente che l’accesso è fallito.

```
else:
    # Se le credenziali non corrispondono a nessun record nel database, informa
    print('<html>')
    print('<head>')
    print('<title>Login Fallito</title>')
    print('<link rel="stylesheet" type="text/css" href="style.css">')
    print('</head>')
    print('<body>')
    print('<h2>Login Fallito! Accesso Negato</h2>')
    print('<div style="text-align:center;"><a href="http://127.0.0.1:90/login.ht
    print('</body>')
    print('</html>')

# Chiusura della connessione
cursor.close()
cnx.close()      # Queste linee chiudono il cursore e la connessione al database
```

Client:

Questo script Python è progettato per gestire la ricerca di immobili in un'applicazione web. Inizialmente, stabilisce una connessione con un database MySQL e recupera i filtri di ricerca inviati dall'utente attraverso un modulo web. Successivamente, costruisce una query SQL in base a questi filtri e la esegue per recuperare i risultati.





Se ci sono risultati, li stampa in una tabella HTML. Ogni riga della tabella rappresenta un immobile e include dettagli come l'ID, l'indirizzo, il numero di vani, la metratura, il piano, la presenza dell'ascensore, il prezzo, lo stato di vendita, il nome dell'agenzia e la località.

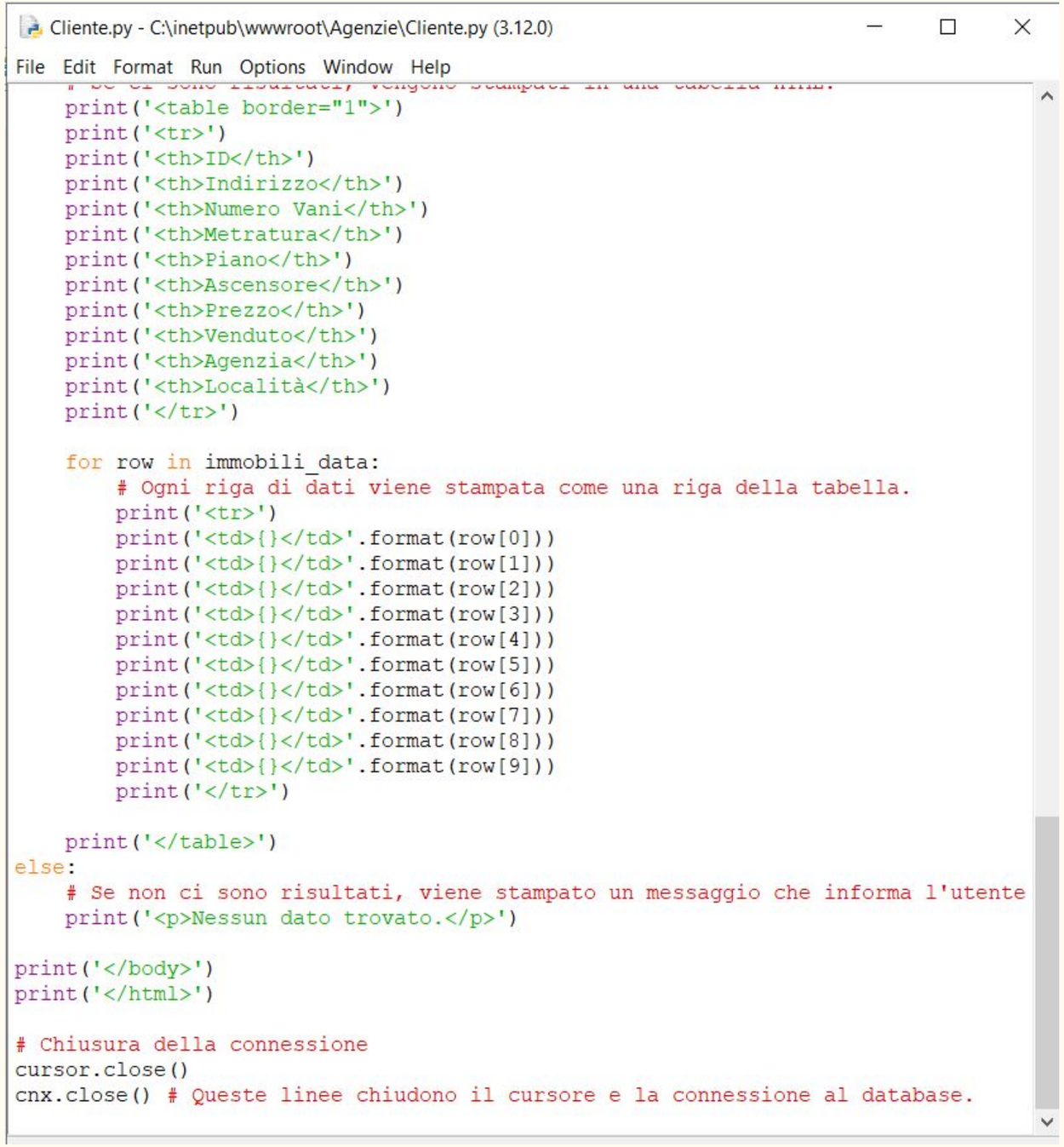
Se non ci sono risultati, viene visualizzato un messaggio che informa l'utente che non sono stati trovati dati. Infine, il cursore e la connessione al database vengono chiusi per liberare le risorse.

Inoltre, il modulo di ricerca viene stampato nuovamente, permettendo all'utente di effettuare una nuova ricerca senza dover ricaricare la pagina. I filtri di ricerca includono la presenza dell'ascensore e la località, e l'utente può selezionare i valori desiderati da un menu a discesa.

Cliente.py - C:\inetpub\wwwroot\Agenzie\Cliente.py (3.12.0)

File Edit Format Run Options Window Help

```
# Stampare la form HTML per la ricerca
print('<html>')
print('<head>')
print('<title>Ricerca Immobili</title>')
print('<link rel="stylesheet" type="text/css" href="style.css">')
print('</head>')
print('<body>')
print('<h2>Ricerca Immobili</h2>')

print('<form method="post" action="">')
print('<label for="ascensore">Ascensore:</label>')
print('<select name="ascensore">')
print('<option value="Tutti" selected>Tutti</option>')
print('<option value="Si" {}>Si</option>'.format('selected' if ascensore_filter
print('<option value="No" {}>No</option>'.format('selected' if ascensore_filter
print('</select>')

print('<label for="localita">Località:</label>')
print('<select name="localita">')
print('<option value="Tutti" selected>Tutti</option>')
print('<option value="Torino" {}>Torino</option>'.format('selected' if localita_
print('<option value="Alessandria" {}>Alessandria</option>'.format('selected' if
print('<option value="Asti" {}>Asti</option>'.format('selected' if localita_filt
print('</select>')

print('<input type="submit" value="Ricerca">')
print('</form>')

print('<h2>Risultati della Query</h2>') # Queste linee stampano un modulo HTML c

# Verifica se ci sono risultati nella ricerca
if immobili_data:
    # Se ci sono risultati, vengono stampati in una tabella HTML.
    print('<table border="1">')
    print('<tr>')
    print('<th>ID</th>')
    print('<th>Indirizzo</th>')
    print('<th>Numero Vani</th>')
    print('<th>Metratura</th>')
    print('<th>Piano</th>')
    print('<th>Ascensore</th>')
```



In sintesi, questo script gestisce la ricerca di immobili in un'applicazione web, permettendo all'utente di filtrare i risultati in base alla presenza dell'ascensore e alla località. I risultati vengono visualizzati in una tabella HTML per una facile lettura, e se non ci sono risultati, viene visualizzato un messaggio appropriato. Infine, il modulo di ricerca viene stampato nuovamente, permettendo all'utente di effettuare una nuova ricerca immediatamente.

Administrator:

Questo codice HTML rappresenta un modulo di inserimento per un amministratore di un'applicazione web di immobili. L'amministratore può inserire dettagli su un immobile, come l'ID, l'indirizzo, il numero di vani, la metratura, il piano, la presenza dell'ascensore, il prezzo, lo stato di vendita, l'agenzia e la località. Una volta compilato il modulo, i dati vengono inviati a un file chiamato "admin.py" sul server per l'elaborazione.

In sintesi, questo modulo consente agli amministratori di inserire nuovi immobili nel database dell'applicazione web.

admin.html - Blocco note

File Modifica Formato Visualizza ?

```
<!DOCTYPE html>
<html lang="it">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" type="text/css" href="style.css">
    <title>Inserimento Immobili</title>
</head>
<body>
    <h2>Inserimento informazioni Amministratore</h2>
    <form action="admin.py" method="post">

        <label for="id">ID:</label>
        <input type="text" id="id" name="id" required><br>

        <label for="indirizzo">Indirizzo:</label>
        <input type="text" id="indirizzo" name="indirizzo" required><br>

        <label for="nvani">Numero Vani:</label>
        <input type="text" id="nvani" name="nvani" required><br>

        <label for="metratura">Metratura:</label>
        <input type="text" id="metratura" name="metratura" required><br>

        <label for="piano">Piano:</label>
        <input type="text" id="piano" name="piano" required><br>

        <label for="ascensore">Ascensore:</label>
        <select id="ascensore" name="ascensore" required>
            <option value="Si">Si</option>
            <option value="No">No</option>
        </select><br>
```

Windows (CRLF) Linea 55, colonna 17 100%

admin.py - C:\inetpub\wwwroot\Agenzie\admin.py (3.12.0)

File Edit Format Run Options Window Help

```
# La libreria cgi viene importata per gestire i dati del modulo inviati dall'utente.
import cgi

# La libreria mysql.connector viene importata per connettersi al database MySQL.
import mysql.connector

# Questa linea stampa l'intestazione HTTP necessaria per un documento HTML.
print('Content-type: text/html\n\n')

# Connessione al database
# Questa linea stabilisce una connessione al database MySQL.
cnx = mysql.connector.connect(user='root', password='1405FilippoSor!',
                              host='127.0.0.1',
                              database='agenzia')

cursor = cnx.cursor() # Questa linea crea un cursore che può essere utilizzato per es

# Form di inserimento dati immobile
form = cgi.FieldStorage() # Questa linea crea un oggetto FieldStorage che contiene i

# Recupero dei dati dal form
id_immobile = form.getvalue('id')
indirizzo = form.getvalue('indirizzo')
nvani = form.getvalue('nvani')
metratura = form.getvalue('metratura')
piano = form.getvalue('piano')
ascensore = form.getvalue('ascensore')
prezzo = form.getvalue('prezzo')
venduto = form.getvalue('venduto')
agenzia_nome = form.getvalue('agenzia')  # Nome dell'agenzia inserito dall'utente
localita_nome = form.getvalue('localita')  # Nome della località inserito dall'utente

# Ottieni l'ID dell'agenzia dalla tabella "agenzia" usando il nome
cerca_agenzia = "SELECT id_agenzia FROM agenzia WHERE Ind = %s"
cursor.execute(cerca_agenzia, (agenzia_nome,))
result_agenzia = cursor.fetchone()
if result_agenzia:
    id_agenzia = result_agenzia[0]
else:
    # Puoi gestire un'agenzia non trovata in base alle tue esigenze
```

Questo script Python gestisce l'inserimento di nuovi immobili in un'applicazione web. Recupera i dati di un immobile inviati dall'amministratore attraverso un modulo web, cerca gli ID corrispondenti per l'agenzia e la località nel database, e poi inserisce il nuovo immobile nel database.

Se l’inserimento è riuscito, viene visualizzato un messaggio di successo. Se si verifica un errore durante l’inserimento, viene visualizzato un messaggio di errore. Infine, il cursore e la connessione al database vengono chiusi per liberare le risorse.

admin.py - C:\inetpub\wwwroot\Agenzie\admin.py (3.12.0)

File Edit Format Run Options Window Help

```
# Ottieni l'ID della località dalla tabella "localita" usando il nome
cerca_localita = "SELECT id_loc FROM localita WHERE nome = %s"
cursor.execute(cerca_localita, (localita_nome,))
result_localita = cursor.fetchone()

# Leggi completamente i risultati di cerca_localita prima di procedere
cursor.fetchall()

if result_localita:
    id_localita = result_localita[0]
else:
    # Puoi gestire una località non trovata in base alle tue esigenze
    id_localita = None

# Esecuzione della query solo se il form è stato inviato e l'azione è "Inserisci"
if form.getvalue('action_inserisci'):
    try:
        # Query di inserimento nella tabella "immobile"
        inserisci_immobile_query = """
            INSERT INTO immobile
            (id_imm, indirizzo, nvani, metratura, piano, ascensore, prezzo, Venduto,
            VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s)
        """
        dati_immobile = (id_immobile, indirizzo, nvani, metratura, piano, ascensore,

        cursor.execute(inserisci_immobile_query, dati_immobile)
        cnx.commit()

        # Dopo l'inserimento, puoi anche effettuare un redirect o mostrare un messagg
        print('<p>Dati inseriti con successo!</p>')
        print('<div style="text-align:center;"><a href="http://127.0.0.1:90/admin.htm
    except mysql.connector.Error as err:
        # Gestisci eventuali errori durante l'inserimento
        print('<p>Errore durante l\'inserimento dei dati: {}</p>'.format(err))
        print('<div style="text-align:center;"><a href="http://127.0.0.1:90/admin.htm

# Chiusura della connessione
cursor.close()
cnx.close()# Queste linee chiudono il cursore e la connessione al database.
```

Registrazione Cliente:

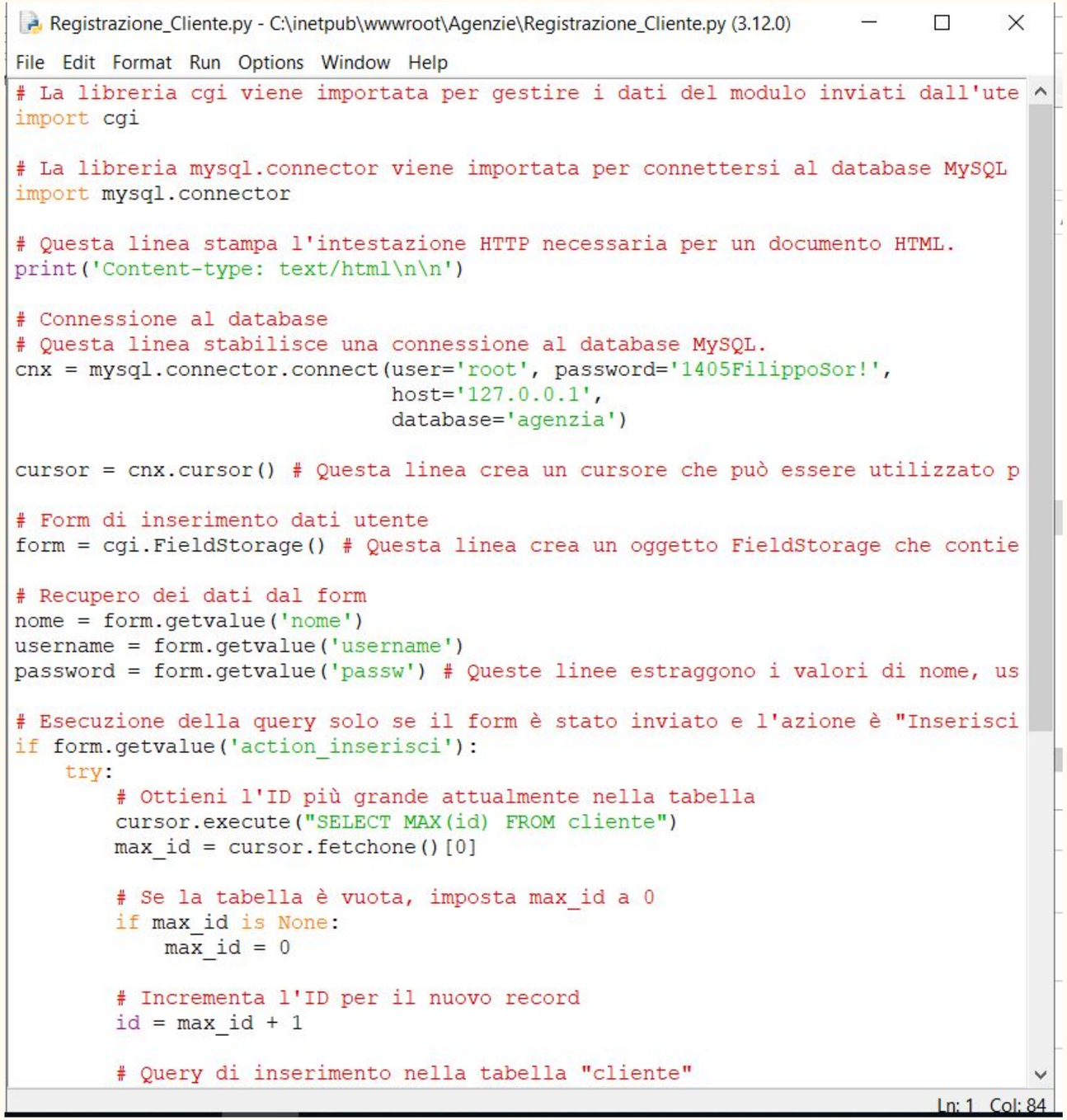
Questo codice HTML rappresenta un modulo di registrazione per un utente in un'applicazione web. L'utente può inserire il proprio nome, username e password. Una volta compilato il modulo, i dati vengono inviati a un file chiamato "Registrazione_Cliente.py" sul server per l'elaborazione. In sintesi, questo modulo consente agli utenti di registrarsi nell'applicazione web.

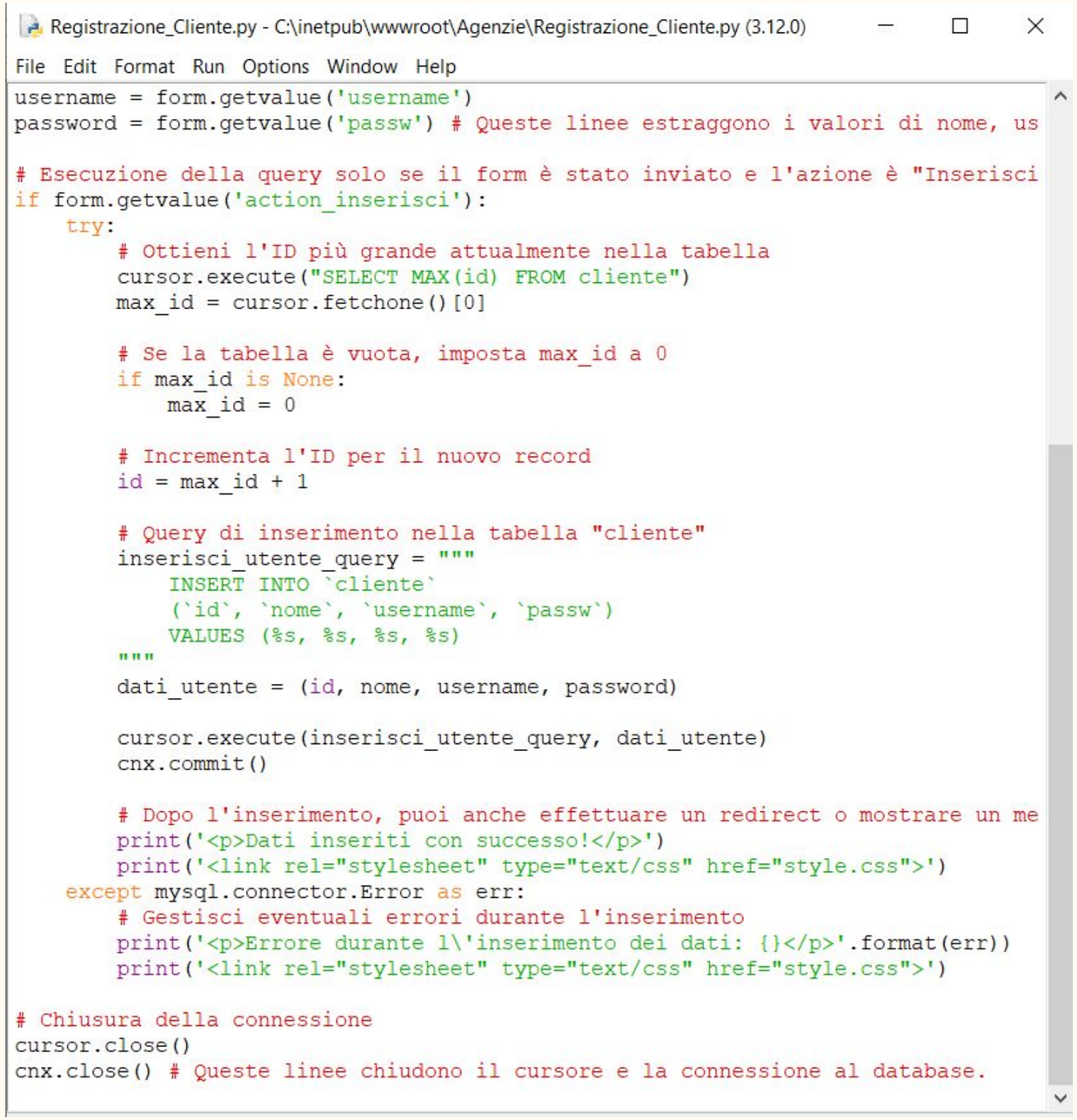
Registrazione_Cliente.html - Blocco note

File Modifica Formato Visualizza ?

```
<!DOCTYPE html>
<html lang="it">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" type="text/css" href="style.css">
    <title>Inserimento Utente</title>
</head>
<body>
    <h2>Inserimento informazioni Utente</h2>
    <form action="Registrazione_Cliente.py" method="post">

        <label for="nome">Nome:</label>
        <input type="text" id="nome" name="nome" required><br>

        <label for="username">Username:</label>
        <input type="text" id="username" name="username" required><br>

        <label for="passw">Password:</label>
        <input type="password" id="passw" name="passw" required><br>

        <input type="submit" value="Inserisci" name="action_inserisci">

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

Questo script Python è progettato per gestire la registrazione di nuovi utenti in un'applicazione web di immobili. Inizialmente, stabilisce una connessione con un database MySQL. Successivamente, recupera i dati dell'utente inviati attraverso un modulo web.
Questi dati includono il nome, l'username
e la password dell'utente.
Il script genera poi un ID unico per il nuovo utente. Questo viene fatto recuperando l'ID più grande attualmente presente nella tabella dei clienti del database e incrementandolo di uno.
Una volta generato l'ID, il nuovo utente viene inserito nel database. Questo viene fatto eseguendo una query SQL di inserimento che aggiunge un nuovo record alla tabella dei clienti con i dati dell'utente e l'ID generato.





Se l'inserimento dei dati nel database è riuscito, il script stampa un messaggio di successo. Se si verifica un errore durante l'inserimento dei dati, il script stampa un messaggio di errore.

Infine, il cursore e la connessione al database vengono chiusi. Questo è importante per liberare le risorse del sistema e prevenire possibili problemi di sicurezza.

In sintesi, questo script gestisce l'intero processo di registrazione di un nuovo utente, dal recupero dei dati dell'utente all'inserimento di questi dati nel database. Gestisce anche eventuali errori che possono verificarsi durante questo processo.

Grazie per l’attenzione