**Progetto 2 - Caso Webservices**

Gruppo **GossipWeb**:

*Lara Longhi (1079261)*

*Federico Carrara (1080669)*

*Alessia Lazzari (1078863)*

*Consegna*:

Con i webservice deve essere possibile migrare i dati che avete inserito su Altervista nel 1° progetto e migrarli verso un database locale su Postgres

Il webservice remoto è consultabile al link: <http://servsanitariopw9.altervista.org/WS.php>

Il webservice locale e la servlet sono visibili al link: <https://github.com/LLonghi02/PW24_WebService>

**Fasi per l’installazione e l’avviamento**

1. **Scaricare il repository da GitHub**: <https://github.com/LLonghi02/PW24_WebService>
   * In questa repository, premere il pulsante code → download ZIP
   * Estrarre il file ZIP
2. **Avviare Apache Tomcat**
   * Comando windows+R → Aprire: services.msc
   * Click col tasto destro su “Apache Tomcat 10.1 Tomcat10” (o simile in base alla versione presente sul sistema) → avvia
   * Copiare la cartella “AppWeb” presente al percorso "C:\...\PW24\_WebService-main\PW24\_SSanitario\Servlet” nella cartella "C:\Program Files\Apache Software Foundation\Tomcat 10.1\webapps" presente nel tuo sistema
3. **Creare un Database PostgreSQL**
   * Aprire “pgAdmin4”
   * Aprire il tuo Server
   * Fare clic con il tasto destro su Databases → Create → Database…
   * Chiamare il Database “ServSanitario” e salva
4. **Inserire i tuoi dati di accesso PostgreSQL in settings.py**
   * Aprire settings.py in VSC (va bene qualsiasi editor di testo), trovabile al percorso "C:\...\PW24\_WebService-main\PW24\_SSanitario\PW24\_SSanitario"
   * Modificare DATABASES alla riga 80 con il tuo USER e la tua PASSWORD

Immagine che contiene testo, schermata, software, schermo

Descrizione generata automaticamente

1. **Aprire il terminale**
   * Aprire il prompt dei comandi (attenzione NON deve essere Windows PowerShell)
2. **Installare le dipendenze e configurare l’ambiente**
   * Copiare i codici presenti nel file step\_SSanitario.ipynb nella console del terminale e seguire le indicazioni
3. **Entrare in Apache Tomcat**
   * Aprire il browser
   * Inserire nella barra di ricerca <http://localhost:8080/AppWeb/servletSanitario>
4. **Visualizzare le tabelle migrate in PostgreSQL**
   * Aprire “pgAdmin4” o effettua un refresh Databases se ancora aperto dal punto 3
   * Aprire il Database “ServSanitario”, creato in precedenza, estendi Schemas → public → Tables

**“Dump” del database di origine**: DB – Ex3: Servizio Sanitario

Immagine che contiene testo, schermata, software, Icona del computer

Descrizione generata automaticamente

|  |  |
| --- | --- |
| Tabella Cittadino Immagine che contiene testo, software, Icona del computer, Software multimediale  Descrizione generata automaticamente | Tabella Ospedale Immagine che contiene testo, schermata, software, Icona del computer  Descrizione generata automaticamente |
| Tabella Patologia Immagine che contiene testo, software, Icona del computer, Software multimediale  Descrizione generata automaticamente | Tabella Ricovero  Immagine che contiene testo, schermata, software, Icona del computer  Descrizione generata automaticamente |

|  |  |  |
| --- | --- | --- |
| Tabella PatologiaCronica  Immagine che contiene testo, software, Icona del computer, Software multimediale  Descrizione generata automaticamente | Tabella PatologiaMortale  Immagine che contiene testo, schermata, software, Icona del computer  Descrizione generata automaticamente | Tabella PatologiaRicovero  Immagine che contiene testo, schermata, software, Icona del computer  Descrizione generata automaticamente |

**Informazioni di progetto**

*Descrizione del progetto*

Il progetto utilizza Django per migrare dati da un server remoto a uno locale. L'architettura prevede l'interazione tra un'applicazione Django e una servlet Java, che opera su un server Tomcat, per effettuare la migrazione dei dati tramite richieste HTTP.

*Scelte di progetto*

Utilizzo di Tomcat

* Server di Servlet Affidabile: Tomcat è uno dei server di servlet più popolari e affidabili.
* Compatibilità e Performance: Tomcat è ottimizzato per eseguire servlet Java e offre buone prestazioni per le applicazioni web basate su Java.
* Configurazione Semplice: La configurazione di servlet e altre risorse web è facilitata tramite il file web.xml, permettendo una gestione centralizzata delle impostazioni dell'applicazione.

*Informazioni sulle tabelle in PostgreSQL*

Django crea automaticamente alcune tabelle necessarie per il funzionamento del sistema. Grazie al file ‘routers.py’, abbiamo evitato la creazione della maggior parte di queste tabelle. Al termine del punto 8, in PostgreSQL potrai vedere le tabelle indicate nel dump e una tabella speciale chiamata ‘django\_migrations’.

La tabella ‘django\_migration’ è gestita direttamente da Django ed è fondamentale per il sistema di migrazioni. Questa tabella è indispensabile e non può essere eliminata, poiché tiene traccia delle migrazioni già applicate e consente a Django di sapere quali migrazioni devono ancora essere eseguite.

*Descrizione file presenti nella cartella* PW24\_SSanitario*:*

1. Applicazione Django: importer

L'applicazione importer è una componente chiave del progetto le sue principali funzionalità includono:

* views.py : Contiene le viste che gestiscono le richieste HTTP e restituiscono le risposte appropriate.
* models.py : Definisce le strutture dei dati e le interazioni con il database.

2. Servlet

La servlet è strutturata in due directory principali: AppWeb e src.

* AppWeb/WEB-INF:
  + web.xml: File di configurazione per la servlet in un'applicazione web Java.
  + classes:
    - Servlet.class: Il file compilato della servlet, generato dal file Servlet.java.
* src:
  + Servlet.java: Classe che implementa la servlet Java con le seguenti funzionalità:
    - doGet e doPost : Gestiscono le richieste HTTP GET e POST, rispettivamente, chiamando il metodo ‘handleRequest’.
    - handleRequest :
    - Recupera dati da un webservice remoto su Altervista con una richiesta HTTP POST.
    - Invia i dati recuperati a un webservice locale Django tramite un'altra richiesta HTTP POST.
    - fetchDataFromRemoteService : Effettua la richiesta al webservice remoto e legge la risposta.
    - sendToDjango : Invia i dati al server Django locale e gestisce la risposta.

Questa servlet facilita la migrazione dei dati tra un server remoto e un'applicazione Django locale. La servlet opera su un server Tomcat, che gestisce l'esecuzione della servlet e le interazioni con

webservice.

3. Progetto Django: PW24\_SSanitario

* urls.py : File principale che mappa gli URL del progetto Django alle relative viste e applicazioni. Questo file coordina la navigazione all'interno del progetto.
* settings.py : Contiene tutte le impostazioni di configurazione per il progetto Django, come la configurazione del database, le applicazioni installate, le impostazioni di sicurezza, e altri parametri necessari per il funzionamento dell'applicazione.
* routers.py : Personalizza la gestione delle operazioni di lettura e scrittura e controlla le migrazioni, escludendo alcune app di sistema dalle migrazioni.