

# MetaDoctor Tracker

Anna-Teresa Arcione (3397875),  
Hithesh Chettenahalli Honnegowda (3745113),  
Julia Jezlow (3412006)

Text Technology

02.07.24

# Motivation

- Finding reliable doctors can be a challenging task, especially in a bustling city like Stuttgart. Our project aims to simplify this process by providing a robust platform where users can easily filter and discover the best doctors in Stuttgart.
- By leveraging metadata and advanced search functionalities, users can refine their search based on specific criteria such as medical specialty, clinic location, patient reviews, and appointment availability.
- This not only saves valuable time but also ensures that users find doctors who meet their exact needs and preferences.

# What is Metadata

- XML Metadata plays an even more important role as it initializes the document model and typically takes the form of the XML document that the XML Schema is based on or defined by the XML Schema.
- This is particularly useful in clarifying the resultant document structures when dealing with nested Schemas. It also set the standard used between different examples for data exchange.

# Project Objectives

- **Aim of this project:** streamline the process of collecting, preparing, and accessing metadata of doctor information from an XML file and inserting it into a PostgreSQL database
- **Main objectives:**
  - Efficient data collection from external XML sources
  - Proper preparation and validation of collected data
  - Reliable access and insertion of data into a PostgreSQL database

# Collect – Data Collection

- Start by collecting data from an external XML file
- XML file includes information like: name of the doctor, speciality, address, contact details and if there is a possibility to make the appointment online
- Tools and libraries used for parsing XML files: xml.etree.ElementTree
- XML code snippet:

```
<?xml version="1.0" encoding="UTF-8"?>
<Doctors>
  <Doctor>
    <ID>1</ID>
    <Name>Hausärzte Stadtmitte</Name>
    <Speciality>general practitioner</Speciality>
    <Address>Marienstraße 28</Address>
    <ClinicOffice>joint practice</ClinicOffice>
    <ZIPCode>70178</ZIPCode>
    <City>Stuttgart</City>
    <PhoneNumber>0711-2200144</PhoneNumber>
    <Website>http://hausarzte-stadtmitte.de/en/</Website>
    <Rating>4.2</Rating>
    <OnlineAppointment>No</OnlineAppointment>
  </Doctor>
```

# Prepare – Data Collection

- Next parse and prepare the data for insertion
- XML data is transformed into a structured format suitable for database insertion
- Code snippet of parsing function:

```
import psycpg2
import xml.etree.ElementTree as ET

# Function to parse the XML file
def parse_xml(file_path):
    tree = ET.parse(file_path)
    root = tree.getroot()
    doctors = []

    for doctor in root.findall('Doctor'):
        name = doctor.find('Name').text
        speciality = doctor.find('Speciality').text
        address = doctor.find('Address').text
        clinic_office = doctor.find('ClinicOffice').text
        zip_code = doctor.find('ZIPCode').text
        city = doctor.find('City').text
        phone_number = doctor.find('PhoneNumber').text
        website = doctor.find('Website').text
        rating = float(doctor.find('Rating').text)
        online_appointment = doctor.find('OnlineAppointment').text.lower() == 'yes'

        doctors.append((name, speciality, address, clinic_office, zip_code, city, phone_number, website, rating, online_appointment))

    return doctors
```

# Access – Data Collection

- connect to the PostgreSQL database and insert the prepared data
- The database schema includes fields such as Name, Speciality, Address, and more
- Below are snippets of the SQL table creation and data insertion code:

```
# Create table if it doesn't exist
create_table_query = '''
CREATE TABLE IF NOT EXISTS Doctors (
    Name VARCHAR(255),
    Speciality VARCHAR(255),
    Address VARCHAR(255),
    ClinicOffice VARCHAR(255),
    ZIPCode VARCHAR(10),
    City VARCHAR(255),
    PhoneNumber VARCHAR(20),
    Website VARCHAR(255),
    Rating FLOAT,
    OnlineAppointment BOOLEAN
)
'''
```

```
# Function to insert data into the PostgreSQL database
def insert_data_to_postgres(doctors):
    try:
        # Connect to the PostgreSQL database
        conn = psycopg2.connect(
            dbname="postgres",
            user="postgres",
            password="0000",
            host="localhost"
        )
        cursor = conn.cursor()

        # Create table if it doesn't exist
        create_table_query = '''
        CREATE TABLE IF NOT EXISTS Doctors (
            Name VARCHAR(255),
            Speciality VARCHAR(255),
            Address VARCHAR(255),
            ClinicOffice VARCHAR(255),
            ZIPCode VARCHAR(10),
            City VARCHAR(255),
            PhoneNumber VARCHAR(20),
            Website VARCHAR(255),
            Rating FLOAT,
            OnlineAppointment BOOLEAN
        )
        '''
        cursor.execute(create_table_query)

        # Insert data into the table
        insert_query = '''
        INSERT INTO Doctors (Name, Speciality, Address, ClinicOffice, ZIPCode, City, PhoneNumber, Website, Rating, OnlineAppointment)
        VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s)
        '''
        cursor.executemany(insert_query, doctors)

        # Commit the transaction
        conn.commit()

        # Close the cursor and connection
        cursor.close()
        conn.close()

        print("Data inserted successfully")
    except Exception as e:
        print(f"Error: {e}")
```

# Accessing desired data for the user

- User can retrieve the data by querying by giving specific condition
  - For example if the user wants to search for doctors with having rating > 4.5 can just run the below query
    - select \* from Doctors where rating>4.5;

```
postgres=# select * from Doctors;
id | name | rating | onlineappointment | speciality | address | clinicoffice | zipcode | city | phonenumber | website
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----
1 | Hausärzte Stadtmitte | 4.2 | f | general practitioner | Marienstraße 28 | joint practice | 70178 | Stuttgart | 0711-2200144 | http://hausaeerzte-stadtmitte.de/en/
2 | Dr. med. Ioannis Kotrotosos | 4.7 | t | ENT medicine | Kronenstraße 19 | joint practice | 70173 | Stuttgart | 0711-2263123 | https://www.hno-praxis-zentrum-stuttgart.de/
3 | Dr. med. Roger Brauchle | 4.8 | f | general practitioner | Tübinger Straße 18 | private practice | 70178 | Stuttgart-Mitte | 0711-6200268 | 
16 | Dr. med. Stefan Wittorf | 4.8 | f | internal medicine | Königstraße 66 | private practice | 70173 | Stuttgart-Mitte | 0711-294800 | https://www.internisten-im-netz.de/aerzte/stuttgart/wittorf/startseite.html
(4 rows)
```

```
postgres=# select * from Doctors where rating>4.5;
id | name | rating | onlineappointment | speciality | address | clinicoffice | zipcode | city | phonenumber | website
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----
2 | Dr. med. Ioannis Kotrotosos | 4.7 | t | ENT medicine | Kronenstraße 19 | joint practice | 70173 | Stuttgart | 0711-2263123 | https://www.hno-praxis-zentrum-stuttgart.de/
3 | Dr. med. Roger Brauchle | 4.8 | f | general practitioner | Tübinger Straße 18 | private practice | 70178 | Stuttgart-Mitte | 0711-6200268 | 
16 | Dr. med. Stefan Wittorf | 4.8 | f | internal medicine | Königstraße 66 | private practice | 70173 | Stuttgart-Mitte | 0711-294800 | https://www.internisten-im-netz.de/aerzte/stuttgart/wittorf/startseite.html
(3 rows)
```



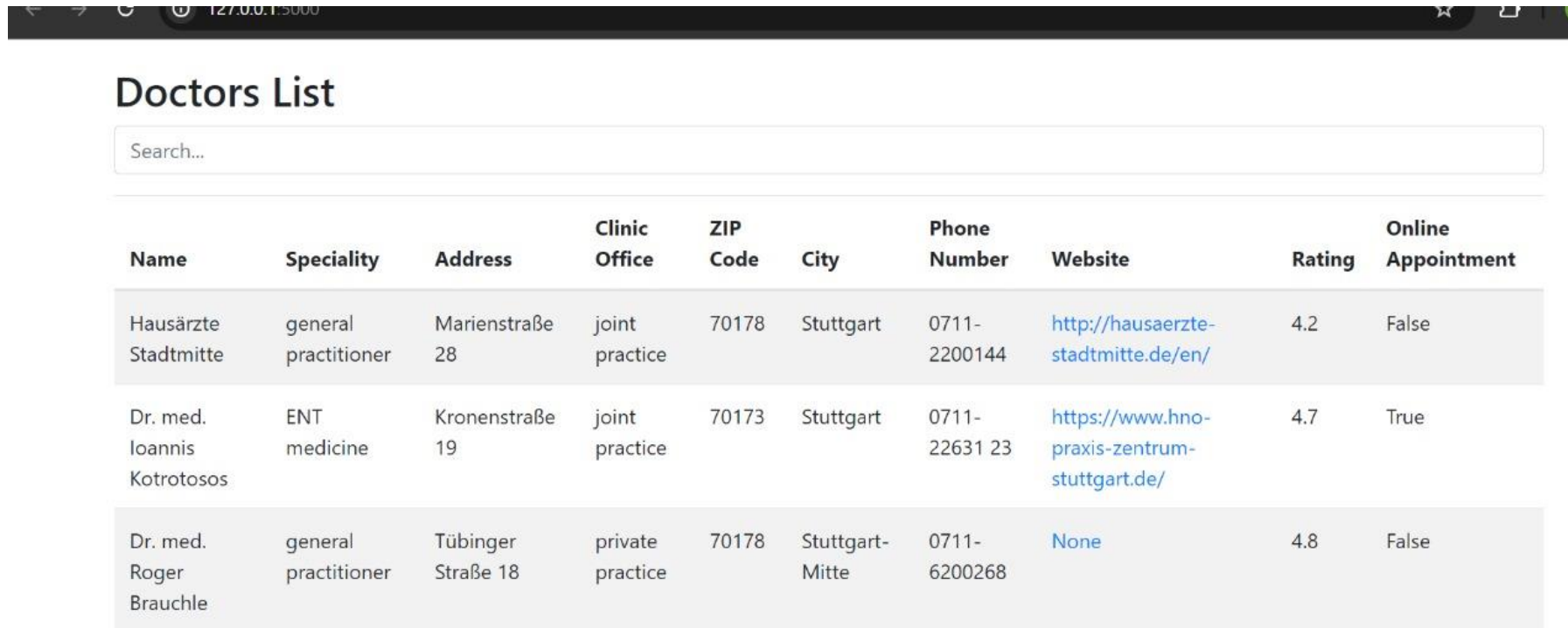
# Implementation – Expected outcomes

- outcomes for each step of the process:
  - Collect: Successful reading and parsing of XML data
  - Prepare: Proper formatting and validation of data
  - Access: Correct insertion of data into PostgreSQL

# Extension - Future Enhancements

- Potential extensions or enhancements to the project include:
  - Adding a web interface for data input
  - Implementing additional data validation and error handling
  - Automating the data collection process with scheduled tasks

# Extension – Web Interface



Search...									
Name	Speciality	Address	Clinic Office	ZIP Code	City	Phone Number	Website	Rating	Online Appointment
Hausärzte Stadtmitte	general practitioner	Marienstraße 28	joint practice	70178	Stuttgart	0711-2200144	<a href="http://hausaeerzte-stadtmitte.de/en/">http://hausaeerzte-stadtmitte.de/en/</a>	4.2	False
Dr. med. Ioannis Kotrotosos	ENT medicine	Kronenstraße 19	joint practice	70173	Stuttgart	0711-22631 23	<a href="https://www.hno-praxis-zentrum-stuttgart.de/">https://www.hno-praxis-zentrum-stuttgart.de/</a>	4.7	True
Dr. med. Roger Brauchle	general practitioner	Tübinger Straße 18	private practice	70178	Stuttgart-Mitte	0711-6200268	None	4.8	False

# Difficulties & Solutions

- Some of the difficulties encountered during the project were:
  - Handling missing or malformed data in the XML file
  - Ensuring secure and stable database connections
  - Dealing with performance issues during bulk data insertion
- Solutions implemented to address these difficulties include:
  - Implemented checks for missing data fields and default values for optional fields
  - Used connection pooling for better database performance
  - Added error handling to manage database connection issues

# Technologies used

- XML
- Postgres SQL
- Python
- HTML
- CSS
- Javascript

# References

- <https://www.psycopg.org/docs/>
- <https://docs.python.org/3/library/xml.etree.elementtree.html>
- <https://www.postgresqltutorial.com/>
- <https://help.nintex.com/>



Thank you  
for your  
attention!

Questions?