

Διαδίκτυο & Εφαρμογές Εξαμηνιαία Εργασία

Κωνσταντína Σκοβολά – 03115112

CTGOV-01

Περιγραφή θέματος

- **CTGOV-01: Αριθμός Ασθενών που συμμετέχουν σε κλινικές μελέτες για μια συγκεκριμένη ασθένεια**
- Η διαδικτυακή εφαρμογή που υλοποιήθηκε χρησιμοποιεί το dataset με όλες τις διαθέσιμες κλινικές δοκιμές του <https://clinicaltrials.gov/> . Η λειτουργία που επιτελεί η εφαρμογή είναι η εύρεση του αριθμού ασθενών που (Α) έχουν ήδη βρεθεί και συμμετέχουν σε κλινική μελέτη σχετική με συγκεκριμένη ασθένεια (type: Actual), (Β) αναμένονται να χρησιμοποιηθούν (type: Anticipated) ή (Γ) έχουν χρησιμοποιηθεί (δεν υπάρχει type) καθώς η κλινική μελέτη έχει ολοκληρωθεί.
- Η υλοποίηση έγινε σε Java (& HTML & CSS). Για τη βάση δεδομένων χρησιμοποιήθηκε η MySQL και για την εισαγωγή των δεδομένων στη βάση, γράφτηκε ένα Python script.

Η βάση δεδομένων (1)

- Τα δεδομένα που έχουμε επιλέξει να κρατήσουμε στη βάση είναι το nct_id, τα mesh terms και ο τύπος του enrollment με τον αντίστοιχο αριθμό, από τα αρχεία XML. Ένα μικρό κομμάτι των δεδομένων στη βάση:

nct_id	mesh_terms	enrollment_type	enrollment_number
NCT00000439	Alcoholism, Bipolar Disorder, Valproic Acid,	Actual	72
NCT00000440	Alcoholism, Naltrexone, Sertraline,	Used	124
NCT00000442	Alcoholism, Naltrexone,	Used	57
NCT00000444	Alcoholism, Nicotine,	Used	500
NCT00000445	Alcoholism, Naltrexone,	Used	300
NCT00000447	Alcoholism, Tobacco Use Disorder, Naltrexone, ...	Used	200
NCT00000448	Alcoholism, Feeding and Eating Disorders, Naltr...	Actual	160
NCT00000449	Alcoholism, Naltrexone,	Used	160
NCT00000450	Alcoholism, Naltrexone,	Actual	159

Η βάση δεδομένων (2)

- Το MySQL script που δημιουργεί τη βάση στο Workbench

```
1 • drop database if exists app_prog;  
2 • create database app_prog;  
3 • use app_prog;  
4  
5 • create table if not exists allxml (  
6     nct_id varchar(255) unique not null,  
7     mesh_terms longtext,  
8     enrollment_type varchar(255) default null,  
9     enrollment_number int default 0  
10 );
```

Η βάση δεδομένων (3)

- Ο κώδικας Python που την εμπλουτίζει με δεδομένα

```
import xml.etree.ElementTree as ET
import mysql.connector
from mysql.connector import Error
import os
import time
import codecs, io
```

Το module ElementTree είναι αυτό που μας επιτρέπει να διαβάσουμε και να διαχειριστούμε τα αρχεία XML. Το κυρίως κομμάτι του κώδικα που κάνει αυτή τη διαχείριση και εισαγωγή των δεδομένων στη βάση ακολουθεί στην επόμενη διαφάνεια

Η βάση δεδομένων (4)

```
32 # do this for all files in the directory AllPublicXML
33 d = 'AllPublicXML'
34 for path, subdirs, files in os.walk(d):
35     for name in files:
36         print(str(os.path.join(path, name)))
37         handle = open(os.path.join(path, name), 'r')
38         try:
39             as_string = handle.read()
40         except UnicodeDecodeError as e:
41             print('Error: ', e)
42             continue
43         f = io.StringIO(as_string)
44         tree = ET.parse(f)
45         root = tree.getroot()
46         mesh_string = ""
47         enrollment_type = None
48         num = 0
49
50         for nct_id in root.iter('nct_id'):
51             nctid = nct_id.text
52
53         for enrollment in root.iter('enrollment'):
54             enrollment_type = "Used"
55             if 'type' in enrollment.attrib:
56                 enrollment_type = enrollment.attrib['type']
57             num = enrollment.text
58
59         for mesh_term in root.iter('mesh_term'):
60             mesh_string += mesh_term.text
61             mesh_string += ', '
62
63         sql = ("INSERT INTO app_prog.allxml "
64              "(nct_id, enrollment_type, enrollment_number, mesh_terms)"
65              "VALUES (%s, %s, %s, %s)")
66         val = (nctid, enrollment_type, num, mesh_string)
67         if enrollment_type or int(num) != 0 :
68             mycursor.execute(sql, val)
```

Η αναζήτηση στη βάση

- Οι σχετικές με την ασθένεια έρευνες είναι αυτές που περιέχουν τον όρο αναζήτησης στα mesh terms τους. Έτσι τα queries μας είναι τα εξής

```
query = "SELECT enrollment_type, sum(enrollment_number) " +  
        "FROM allxml " +  
        "WHERE mesh_terms like " + "'" + condition + "%' " +  
        "GROUP BY enrollment_type";
```

```
query = "SELECT count(*) from allxml where mesh_terms like "+  
        "'" + condition + "%'";
```

Η υλοποίηση σε Java

- Έχουμε δημιουργήσει δύο κλάσεις, το Java Servlet `EnterConditionServlet` και την κλάση `ConditionService`, τη συνάρτηση `getPatientTypes` της οποίας καλεί το Servlet.
- Η συνάρτηση `getPatientTypes(String condition)` δέχεται ως όρισμα τον όρο αναζήτησης του χρήστη, κάνει τα queries στη βάση και επιστρέφει ως αποτέλεσμα ένα «λεξικό», με keys τα `enrollment_types` και το συνολικό αριθμό των σχετικών ερευνών, και values τις αντίστοιχες τιμές

Η κλάση ConditionService

```
28      Connection connection = null;
29      PreparedStatement statement = null;
30      ResultSet rs = null;
31      HashMap<String,Integer> patientTypes = new HashMap<String, Integer>();
32      String query;
33      try {
34          connection = DriverManager.getConnection(url, username, password);
35
36          query = "SELECT enrollment_type, sum(enrollment_number) " +
37                  "FROM allxml " +
38                  "WHERE mesh_terms like " + "'" + condition + "%' " +
39                  "GROUP BY enrollment_type";
40          statement = connection.prepareStatement(query);
41          rs = statement.executeQuery();
42          // get types - # patients
43          while (rs.next()) {
44              patientTypes.put(rs.getString(1), rs.getInt(2));
45          }
46          // get total studies referenced
47          if (rs != null) rs.close();
48          if (statement != null) statement.close();
49
50          query = "SELECT count(*) from allxml where mesh_terms like " +
51                  "'" + condition + "%'";
52          statement = connection.prepareStatement(query);
53          rs = statement.executeQuery();
54          while (rs.next()) {
55              patientTypes.put("Total_studies", rs.getInt(1));
56          }
57      }
58      finally {
59          if (rs != null) rs.close();
60          if (statement != null) statement.close();
61          if (connection != null) connection.close();
62      }
63      return patientTypes;
```

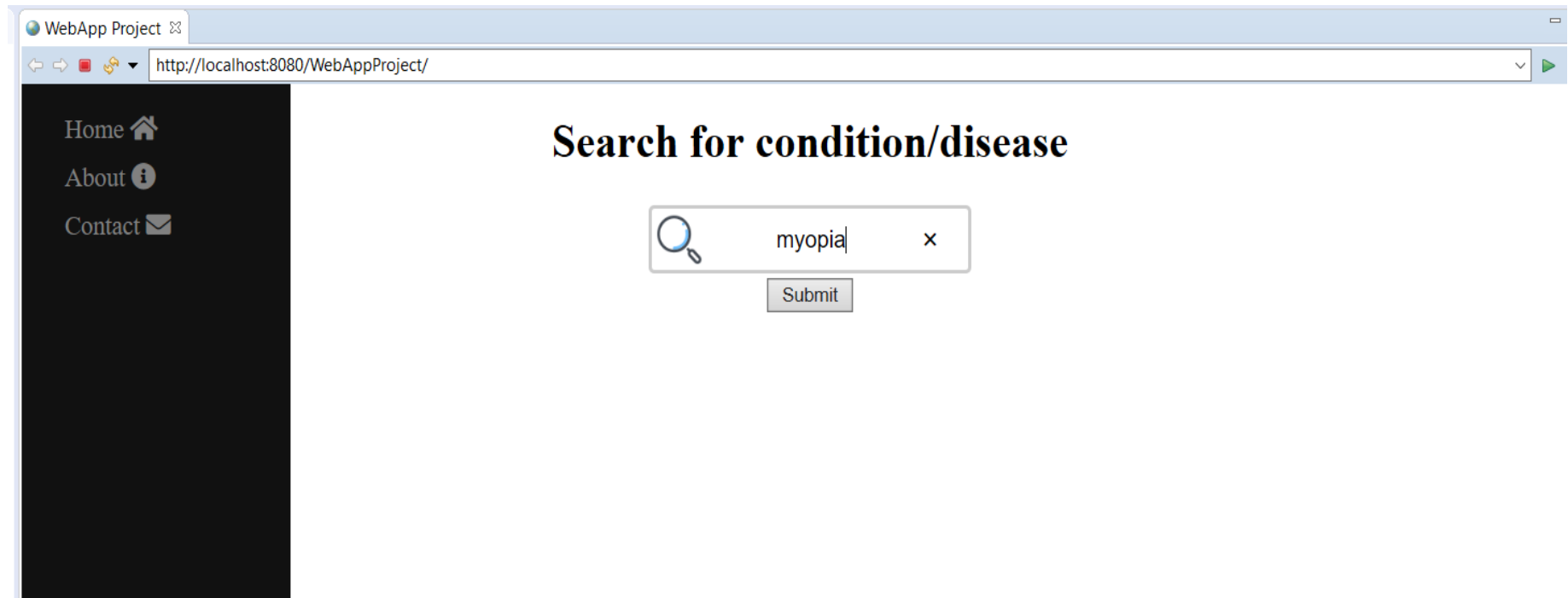
Η κλάση EnterConditionServlet

```
13  @WebServlet(  
14      name="enterconditionervlet",  
15      urlPatterns={"/search"}  
16  )  
17  
18  public class EnterConditionServlet extends HttpServlet{  
19      private static final long serialVersionUID = 1L;  
20      @Override  
21      protected void doPost(HttpServletRequest req, HttpServletResponse resp)  
22      throws ServletException, IOException {  
23          HashMap<String, Integer> patientTypes;  
24          String conditionName = req.getParameter("name");  
25          System.out.print(conditionName);  
26          try {  
27              ConditionService conditionService = new ConditionService();  
28              patientTypes = conditionService.getPatientTypes(conditionName);  
29              req.setAttribute("patientTypes", patientTypes);  
30              req.setAttribute("searchTerm", conditionName);  
31              RequestDispatcher view = req.getRequestDispatcher("results.jsp");  
32              view.forward(req, resp);  
33          }  
34          catch (Exception e){  
35              e.printStackTrace();  
36          }  
37      }  
38  
39      protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
40          response.getWriter().append("Served at ").append(request.getContextPath());  
41  
42          RequestDispatcher dispatcher = request.getRequestDispatcher("index.jsp");  
43          dispatcher.forward(request, response);  
44      }  
45  }
```

Το αρχείο index.jsp

```
1  <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2    pageEncoding="ISO-8859-1"%>
3  <!DOCTYPE html>
4  <html>
5  <head>
6    <meta charset="ISO-8859-1">
7    <title>WebApp Project</title>
8    <link rel="StyleSheet" href="site.css" type="text/css" />
9    <script src="https://kit.fontawesome.com/a076d05399.js"></script>
10  </head>
11  <body>
12
13    <div class="sidenav">
14      <a href="#" > Home <i class="fas fa-home"></i></a>
15      <a href="about.html" > About <i class="far fa-address-card"></i></a>
16      <a href="contact.html"> Contact <i class="fas fa-envelope"></i></a>
17    </div>
18
19    <div align="center" >
20      <h1>Search for condition/disease</h1>
21      <form action="<%= request.getContextPath() %>/search" method="post">
22        <table style="width: 80%">
23          <tr>
24            <td><input type="text" class="search" name="name" placeholder="Search condition..." /></td>
25          </tr>
26        </table>
27        <input type="submit" value="Submit" />
28      </form>
29    </div>
30  </body>
31 </html>
```

Παράδειγμα αναζήτησης



Αποτελέσματα

WebApp Project	
http://localhost:8080/WebAppProject/search	
<h2>Search results for myopia</h2>	
Showing results from 433 relevant studies	
Patient type	Number of patients
Anticipated	205924
Used	1461
Actual	70461

End Notes

- Αναλυτικότερη παρουσίαση κώδικα και επίδειξη των λειτουργιών της εφαρμογής γίνεται στο βίντεο.
- Σημειώνεται ότι η περισσότερη CSS έχει προέλθει από τη σελίδα <https://www.w3schools.com>