



Διαδίκτυο & Εφαρμογές  
Εαρινό Εξάμηνο 2020

# CTGOV-03 Project

Φοιτητής: Ιωάννης Παπίας 03116058

Υπεύθυνος: Ευθύμιος Χονδρογιάννης





## ΘΕΜΑ ΕΡΓΑΣΙΑΣ

Η διαδικτυακή εφαρμογή που υλοποιήθηκε εμφανίζει τον τίτλο των κλινικών μελετών που έχουν λάβει χώρα στην Ελλάδα για μία συγκεκριμένη ασθένεια, καθώς επίσης και τα νοσοκομεία, ινστιτούτα, ιδρύματα κ.λπ. τα οποία συμμετείχαν στις μελέτες, όπως και τις πόλεις στις οποίες αυτά εδρεύουν. Ακόμη, ως επιπλέον πληροφορία δίνεται το status των μελετών, δηλαδή αν έχουν ολοκληρωθεί ή αν βρίσκονται σε φάση εξέλιξης. Τέλος, το πλήθος των κλινικών μελετών ανά πόλη παρουσιάζεται σε χάρτη.

- **Input:** Είναι το όνομα της ασθένειας το οποίο ορίζει ο χρήστης μέσω της web σελίδας.
- **Output 1:** Ο κωδικός κλινικής μελέτης, η πόλη όπου έλαβε χώρα, ο τίτλος, το ίδρυμα – ινστιτούτο το οποίο συμμετείχε και το status της μελέτης.
- **Output 2:** Οι πόλεις ταξινομημένες ανάλογα με τον αριθμό των κλινικών μελετών που έχουν λάβει χώρα.
- **Output 3:** Εμφάνιση των αποτελεσμάτων του output 2 σε χάρτη.



## ΤΕΧΝΟΛΟΓΙΕΣ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΑΝ

Η διαδικτυακή εφαρμογή υλοποιήθηκε με open source τεχνολογίες και συγκεκριμένα περιβάλλον ανάπτυξης *Eclipse IDE* και βάση δεδομένων *MySQL*. Για τις ανάγκες της εφαρμογής έγινε download των κλινικών μελετών που αφορούν την Ελλάδα σε format CSV από τη σελίδα <https://clinicaltrials.gov/>, οι οποίες στην συνέχεια έγιναν import στη βάση. Για την απεικόνιση των ελληνικών πόλεων στις οποίες έλαβαν χώρα κλινικές μελέτες, δημιουργήθηκε πίνακας με πεδία: πόλη, γεωγραφικό πλάτος, γεωγραφικό μήκος. Οι γλώσσες προγραμματισμού που χρησιμοποιήθηκαν για την ανάπτυξη του λογισμικού της εφαρμογής είναι Java, JavaScript, HTML και CSS.

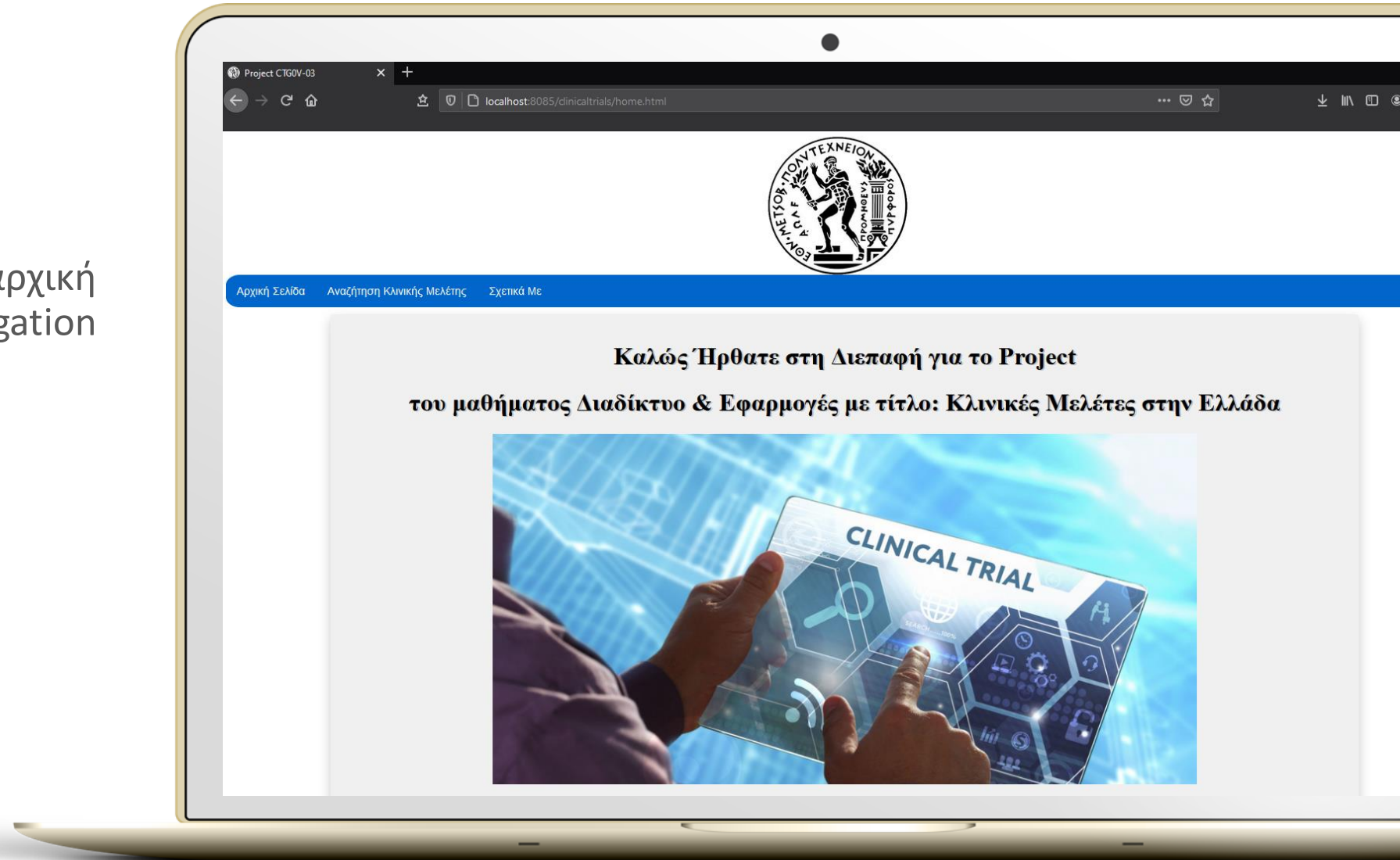


# ΠΑΡΟΥΣΙΑΣΗ ΕΦΑΡΜΟΓΗΣ

## Αρχική Σελίδα

Ο χρήστης εισέρχεται στην αρχική σελίδα της εφαρμογής. Στο navigation bar εμφανίζονται οι επιλογές:

- Αρχική Σελίδα
- Αναζήτηση Κλινικής Μελέτης
- Σχετικά με



# ΠΑΡΟΥΣΙΑΣΗ ΕΦΑΡΜΟΓΗΣ

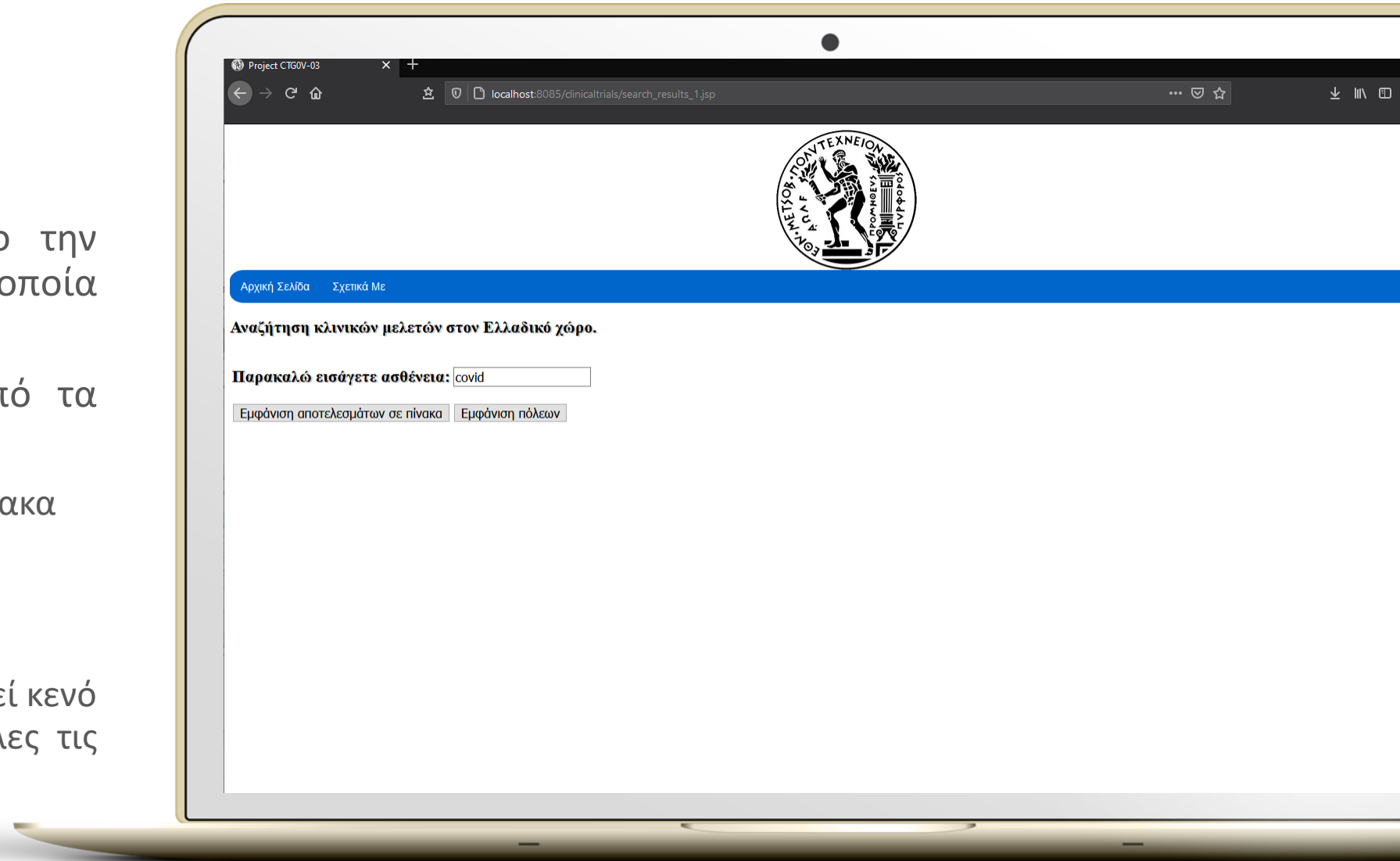
## Αναζήτηση Κλινικής Μελέτης

Ο χρήστης εισάγει στο πεδίο την ασθένεια\*, π.χ. covid, για την οποία επιθυμεί να λάβει πληροφορίες.

Στη συνέχεια επιλέγει ένα από τα ακόλουθα:

- Εμφάνιση αποτελεσμάτων σε πίνακα
- Εμφάνιση πόλεων

\* Σε περίπτωση που το πεδίο αφεθεί κενό οι πληροφορίες αντλούνται για όλες τις ασθένειες.



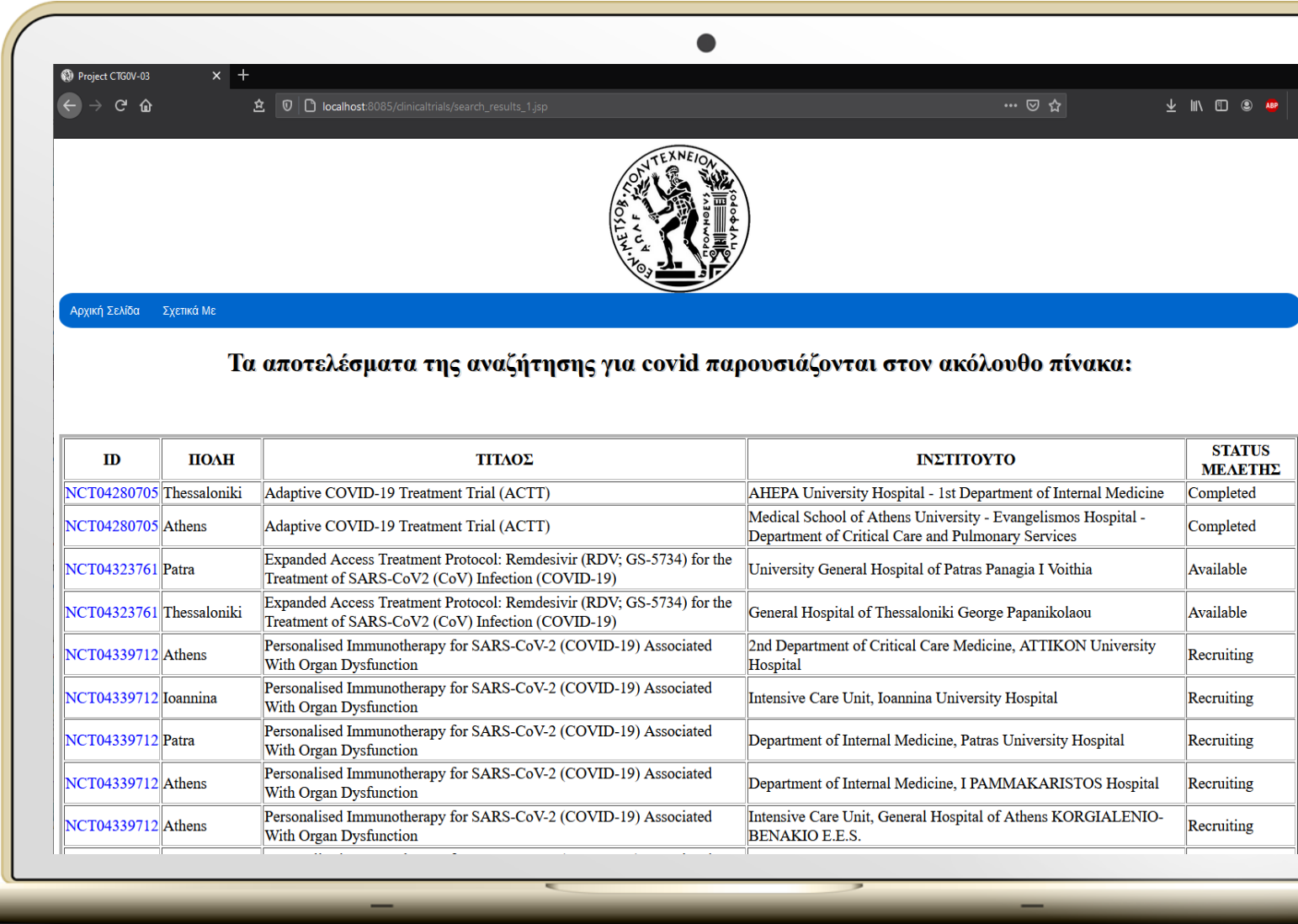
# ΠΑΡΟΥΣΙΑΣΗ ΕΦΑΡΜΟΓΗΣ

## Εμφάνιση αποτελεσμάτων σε πίνακα

Τα αποτελέσματα της αναζήτησης εμφανίζονται σε μορφή πίνακα. Συγκεκριμένα απεικονίζονται:

- ID: Κωδικός κλινικής μελέτης ο οποίος αποτελεί μοναδικό χαρακτηριστικό της κάθε μελέτης. Το πεδίο είναι υπερσύνδεσμος π.χ. <https://clinicaltrials.gov/ct2/show/NCT04280705> και οδηγεί στην ιστοσελίδα clinical trials για περισσότερες πληροφορίες.
- Πόλη: Τοποθεσία - πόλη στην οποία έλαβε χώρα η μελέτη.
- Τίτλος: Τίτλος της μελέτης.
- Ινστιτούτο: Ινστιτούτο, Νοσοκομείο, Εκπαιδευτικό Ίδρυμα που συμμετείχε στη μελέτη.
- Status Μελέτης: Η φάση στην οποία βρίσκεται η μελέτη (completed, available, recruiting κ.λπ.)

Κλινικές μελέτες οι οποίες έλαβαν χώρα σε πολλές πόλεις είτε συμμετείχαν σε αυτές περισσότερα από ένα ινστιτούτα εμφανίζονται με πολλαπλές εγγραφές και ίδιο ID.

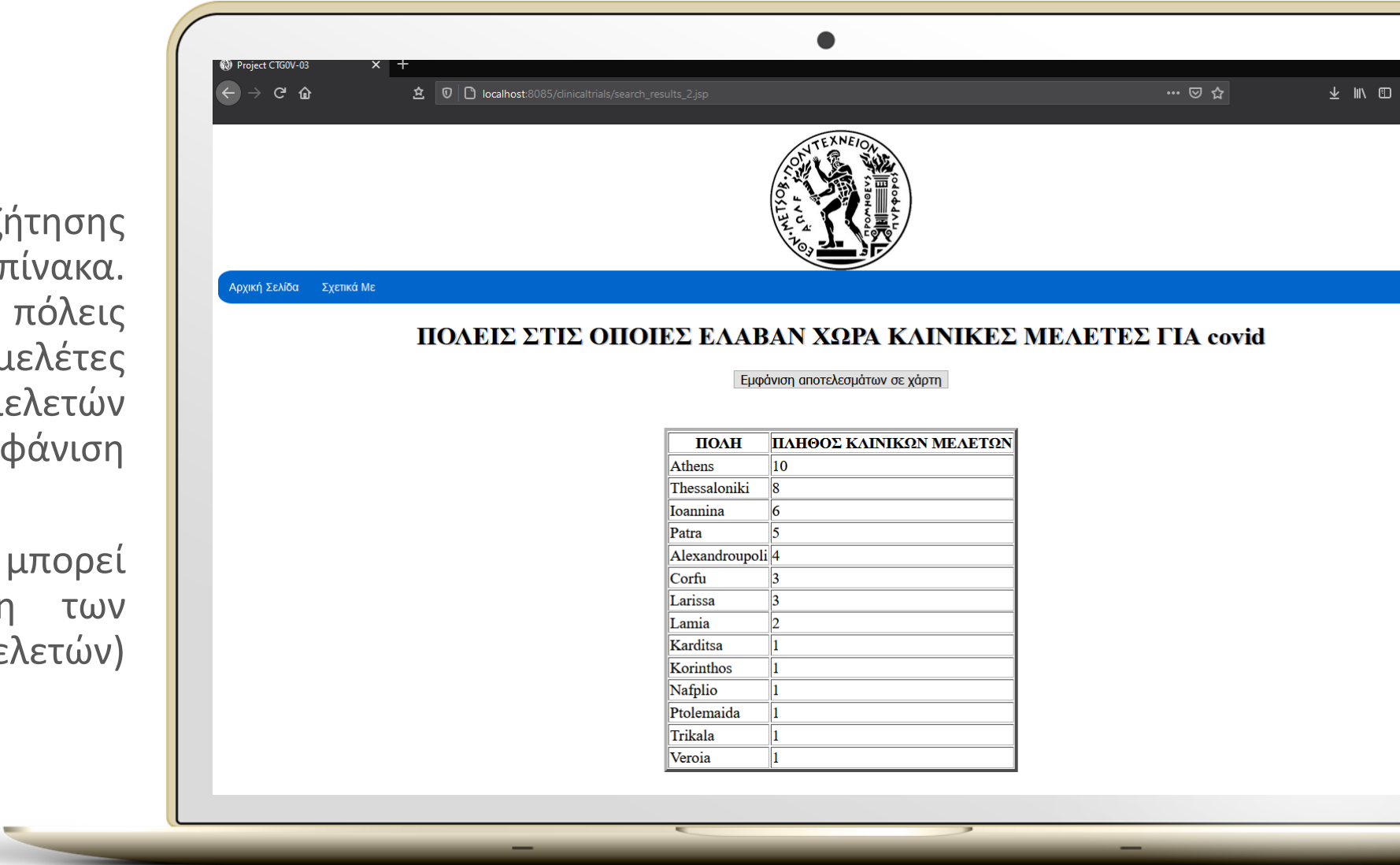


# ΠΑΡΟΥΣΙΑΣΗ ΕΦΑΡΜΟΓΗΣ

## Εμφάνιση πόλεων

Τα αποτελέσματα της αναζήτησης εμφανίζονται σε μορφή πίνακα. Συγκεκριμένα, απεικονίζονται οι πόλεις που συμμετείχαν σε κλινικές μελέτες και το πλήθος των κλινικών μελετών που έλαβαν χώρα σε αυτές. Η εμφάνιση γίνεται με φθίνουσα σειρά.

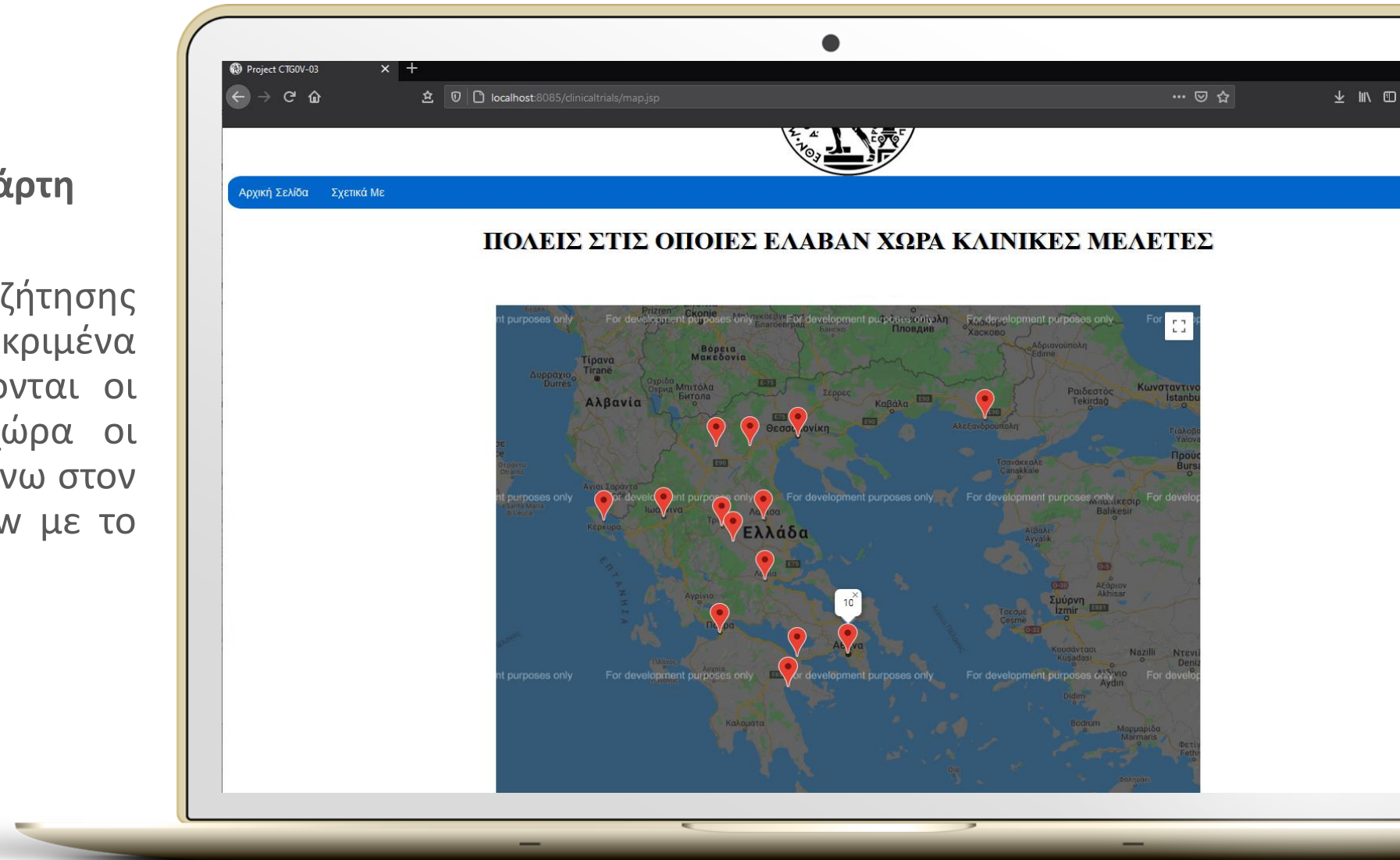
Στην διπλανή οθόνη ο χρήστης μπορεί να επιλέξει την παρουσίαση των αποτελεσμάτων (πόλη-πλήθος μελετών) σε χάρτη.



# ΠΑΡΟΥΣΙΑΣΗ ΕΦΑΡΜΟΓΗΣ

## Εμφάνιση αποτελεσμάτων σε χάρτη

Τα αποτελέσματα της αναζήτησης παρουσιάζονται σε χάρτη. Συγκεκριμένα με κόκκινα markers απεικονίζονται οι πόλεις στις οποίες έλαβαν χώρα οι κλινικές μελέτες, ενώ με κλικ πάνω στον marker εμφανίζεται info-window με το πλήθος των κλινικών μελετών.





# ΠΑΡΟΥΣΙΑΣΗ ΚΩΔΙΚΑ

## Αρχική Σελίδα – [home.html](#)

Η αρχική σελίδα της εφαρμογής, **home.html**, υλοποιήθηκε σε γλώσσα HTML.

Η **HTML** είναι η βασική γλώσσα για δόμηση σελίδων σε περιβάλλον web. Η γλώσσα χρησιμοποιεί αριθμό από tags για την μορφοποίηση κειμένου, για την δημιουργία συνδέσμων – link – μετάβασης ανάμεσα σε σελίδες, για την εισαγωγή εικόνων, για την δημιουργία – εμφάνιση – πινάκων, για την εισαγωγή πληροφορίας από τον χρήστη κ.λπ.

Στη παρούσα εργασία έχει γίνει χρήση της HTML για την μορφοποίηση του front-end και την αλληλεπίδραση με τον χρήστη.

```
home.html x
1 <!DOCTYPE html>
2 <html>
3 <!-- ----- ΑΡΧΙΚΗ ΣΕΛΙΔΑ ----- -->
4 <style>
5 img {
6   display: block;
7   margin-left: auto;
8   margin-right: auto;
9 }
10 </style>
11
12 <head>
13   <title lang='gr'>Project CTG0V-03</title>
14   <meta name='viewport' content='width=device-width, initial-scale=1.0'></meta>
15   <link rel='icon' href='prometheus.png' type='image/png'/>
16
17   <meta name='author' content='Γιάννης Πατιάς'></meta>
18
19   <link rel='stylesheet' href='common.css'></link>
20 </head>
21
22 <body>
23
24   <a href="home.html"></a>
25 <div class="navbar" id="navbar">
26   <a href="home.html">Αρχική Σελίδα</a>
27   <a href="search_results_1.jsp">Αναζήτηση Κλινικής Μελέτης</a>
28   <a href="about.html">Σχετικά Με</a>
29 </div>
30
31 <br/>
32 <br/>
33
34 <div class="main-content">
35
36 <div id='welcome'>
37
38   <h1 style="text-align: center;">Καλώς ήρθατε στη Διεπαφή για το Project</h1>
39   <h1 style="text-align: center;">του μαθήματος Διαδίκτυο & Εφαρμογές με τίτλο: Κλινικές Μελέτες στην Ελλάδα</h1>
40 </div>
41
42   </img>
43
44   <br/>
45   <div style="text-align: right;" class='text'>ΗΜΜΥ ΕΜΠ, Εαρινό Εξάμηνο 2020</div>
46 </div>
47 </body>
48
49 </html>
```

# ΠΑΡΟΥΣΙΑΣΗ ΚΩΔΙΚΑ

## CSS – common.css

Τα CSS (Cascading Style Sheets) καθορίζουν την παρουσίαση των στοιχείων της HTML και διευκολύνουν τον προγραμματισμό των σελίδων.

Στην εφαρμογή γίνεται χρήση του common.css για το style – εμφάνιση – των σελίδων.

```
common.css X
2 *{
3   box-sizing: border-box;
4 }
5
6 body{
7   background-image: url("");
8   background-color: white;
9   background-size: 100%;
10  background-attachment: fixed;
11  display: flex;
12  flex-direction: column;
13  min-height: 100vh;
14 }
15
16 /* Dropdown CSS from w3schools */
17
18 /* Navbar container */
19 .navbar {
20   overflow: hidden;
21   background-color: rgb(0, 102, 204);
22   font-family: Arial;
23   border-radius: 20px / 15px;
24   z-index: 1;
25 }
26
27 /* Links inside the navbar */
28 .navbar a {
29   float: left;
30   font-size: 16px;
31   color: white;
32   text-align: center;
33   padding: 14px 16px;
34   text-decoration: none;
35 }
36
37
38 /* Add a red background color to navbar links on hover */
39 .navbar a:hover{
40   color: white;
41   background-color: rgb(0, 102, 204);
42 }
43
44
45 .main-content {
46   background-color: rgb(240,240,240);
47   background-image: none;
48   max-width: 84%;
49   flex: 1;
50   color: black;
51   box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
52   margin: 10px;
53   padding: 15px;
54   position: relative;
```

# ΠΑΡΟΥΣΙΑΣΗ ΚΩΔΙΚΑ

## Σύνδεση με βάση δεδομένων – ConnectDB.java

Τα Servlets όπως όλα τα προγράμματα Java μπορούν να συνδεθούν με βάσεις δεδομένων με τη χρήση drivers και του JDBC API της Java.

Στην εφαρμογή υλοποιήθηκε το ConnectDB.java για σύνδεση με τη βάση MySQL.

Η κλάση που δημιουργήθηκε καλείται μέσα από τους κώδικες:

- search\_results\_1.jsp
- search\_results\_2.jsp

που παρουσιάζονται στη συνέχεια.

```
ConnectDB.java x
1 package clinicaltrials;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.SQLException;
6
7
8 // Connection to mySQL
9
10 public class ConnectDB {
11
12
13     private static String connectionURL = "jdbc:mysql://localhost:3306/clinicaltrials?useLegacyDatetimeCode=false&serverTimezone=UTC";
14     private static String driverName = "com.mysql.jdbc.Driver";
15     private static String username = "root1";
16     private static String password = "root";
17     private static Connection connection;
18
19
20     public static Connection getConnection() {
21         try {
22             // Load the JDBC driver
23             Class.forName(driverName);
24             try {
25                 // Create a connection to the database
26                 connection = DriverManager.getConnection(connectionURL, username, password);
27             } catch (SQLException ex) {
28                 // Could not connect to the database
29                 System.out.println("Failed to create the database connection.");
30             }
31         } catch (ClassNotFoundException ex) {
32             // Could not find the database driver
33             System.out.println("Driver not found.");
34         }
35         return connection;
36     }
37 }
```

# ΠΑΡΟΥΣΙΑΣΗ ΚΩΔΙΚΑ

Αλληλεπίδραση με το χρήστη για εισαγωγή πληροφορίας και εμφάνιση αποτελεσμάτων – **search\_results\_1.jsp**

Για την υλοποίηση της αλληλεπίδρασης με το χρήστη δημιουργήθηκαν δύο κώδικες JSP. Ο κώδικας **search\_results\_1.jsp** εμφανίζει στον χρήστη το πεδίο (γραμμή 53) στο οποίο εισάγει την ασθένεια\* και ανάλογα με την επιλογή\* του (γραμμή 57 – 58):

- Εγκαθίσταται η σύνδεση με τη MySQL και αντλούνται από τη βάση κλινικές μελέτες οι οποίες περιέχουν στον τίτλο την καταχωρημένη ασθένεια (τμήμα κώδικα Java της μορφής `<% [κώδικας java] %>`) και η εμφάνιση τους θα γίνει σε μορφή πίνακα (κώδικας HTML).
- Είτε θα γίνει ανακατεύθυνση στη σελίδα **search\_results\_2.jsp** (γραμμή 58).

```
search_results_1.jsp
34 // Εισαγωγή ασθένειας για αναζήτηση στη βάση
35 //This method is used to get the value of a request's parameter
36
37 if (request.getParameter("title") == null
38 ) {
39
40 }
41
42 <br />
43 <h2>Αναζήτηση κλινικών μελετών στον Ελλαδικό χώρο.</h2>
44
45 <!-- The POST method is the method the browser uses to talk to the server when asking for a response
46 that takes into account the data provided in the body of the HTTP request.-->
47
48 <form method="post" action="search_results_1.jsp">
49
50 <table>
51 <tr>
52 <td><h2>Παρακαλώ εισάγετε ασθένεια:</h2></td>
53 <td><input type="text" name="title" style="font-size:18px" /></td>
54 </tr>
55
56 <tr>
57 <td colspan="1"><input type="submit" name="submit1" value="Εμφάνιση αποτελεσμάτων σε πίνακα" style="font-size:18px"/></td>
58 <td colspan="1"><input type="submit" name="submit2" value="Εμφάνιση πόλεων" formaction="search_results_2.jsp" style="font-size:18px"/></td>
59 </tr>
60 </table>
61
62 </form>
63
64 <%
65 }else {
66
67 // Σύνδεση με τη βάση και αναζήτηση εγγράφων που ικανοποιούν τη συνθήκη σύμφωνα με την οποία η συγκεκριμένη ασθένεια - λέξη
68 // περιέχεται στον τίτλο της μελέτης
69
70 Connection connection = null;
71 Statement statement = null;
72 ResultSet rs = null;
73
74 String title = request.getParameter("title");
75
76 connection = ConnectDB.getConnection();
77 statement = connection.createStatement();
78
79 rs = statement.executeQuery("Select studies.nct_id, city1 as city, studies.status, title, sponsor_collaborators, name, URL "+
80 " from studies, location, cities "+
81 " where studies.nct_id = location.nct_id and "+
82 " location.city1 = cities.city and "+
83 " title like '%" + title + "%' order by nct_id ");
84
85 if (title == "") {
86 title = "una_Alex_επε_μελέτες";
87 }
```

\*Το tag **form** οριοθετεί την αρχή της φόρμας. Το **action** ορίζει το script το οποίο θα λάβει τα εισαχθέντα δεδομένα (ασθένεια) και η μέθοδος μετάδοσης είναι **method="post"**, η οποία στέλνει όλη την πληροφορία ξεχωριστά από τη URL.



# ΠΑΡΟΥΣΙΑΣΗ ΚΩΔΙΚΑ

Αλληλεπίδραση με το χρήστη για εισαγωγή πληροφορίας και εμφάνιση αποτελεσμάτων (I) – [search\\_results\\_2.jsp](#)

Ο κώδικας `search_results_2.jsp` εγκαθιστά τη σύνδεση με τη MySQL και αντλεί από τη βάση τις πόλεις στις οποίες έχουν λάβει χώρα κλινικές μελέτες για την καταχωρημένη ασθένεια, καθώς και το πλήθος τους (τμήμα κώδικα Java της μορφής `<% [κώδικας java]%>`). Η εμφάνιση τους θα γίνει σε μορφή πίνακα (κώδικας HTML).

Εάν ο χρήστης επιλέξει την «Εμφάνιση αποτελεσμάτων σε χάρτη» γίνεται μετάβαση στη σελίδα `map.jsp`

```
search_results_2.jsp x
49 Statement statement = null;
50 ResultSet rs = null;
51
52 connection = ConnectDB.getConnection();
53 statement = connection.createStatement();
54
55 rs = statement.executeQuery("Select city1 as city , count(DISTINCT studies.nct_id) as num_studies,latitude ,longitude" +
56                             " from studies, location, cities" +
57                             " where studies.nct_id = location.nct_id and" +
58                             " location.city1 = cities.city and " +
59                             " title like '%" + title + "%'" +
60                             " group by city1 order by num_studies desc");
61
62 JSONArray marker = new JSONArray();
63 JSONObject obj = new JSONObject();
64
65 if (title == "") {
66     title = "";
67 } else {
68     title = "ΓΙΑ " + title;
69 }
70
71 %>
72 <center>
73
74 <h1> ΠΟΛΕΙΣ ΣΤΙΣ ΟΠΟΙΕΣ ΕΛΑΒΑΝ ΧΩΡΑ ΚΛΙΝΙΚΕΣ ΜΕΛΕΤΕΣ <%=title%></h1>
75
76
77 <form method="post" action="map.jsp">
78
79     <table>
80     <tr>
81         <td colspan=1><input type=submit value="Εμφάνιση αποτελεσμάτων σε χάρτη" style="font-size:18px"/></td>
82     </tr>
83     </table>
84
85 </form>
86
87
88
89 <br/>
90 <br/>
91
92 <table border="5">
93
94     <tr>
95         <th>ΠΟΛΗ</th>
96         <th>ΠΛΗΘΟΣ ΚΛΙΝΙΚΩΝ ΜΕΛΕΤΩΝ</th>
97     </tr>
98
99     <%
100     int i=0;
101     while (rs.next()) {
```

# ΠΑΡΟΥΣΙΑΣΗ ΚΩΔΙΚΑ

Αλληλεπίδραση με το χρήστη για εισαγωγή πληροφορίας και εμφάνιση αποτελεσμάτων (II) – [search\\_results\\_2.jsp](#)

Εκτός από τις πόλεις και το πλήθος των κλινικών μελετών, από τη βάση αντλούνται επίσης οι συντεταγμένες των πόλεων. Τα παραπάνω αποθηκεύονται σε έναν πίνακα της μορφής **JSON** με όνομα αρχείου **marker.json**. Ο κώδικας για τη δημιουργία του JSON πίνακα και την αποθήκευση σε αρχείο παρουσιάζεται στη διπλανή εικόνα (περισσότερες πληροφορίες στο βίντεο).

Το JSON, είναι ένα ανοικτό πρότυπο που χρησιμοποιεί κείμενο αναγνώσιμο από τον άνθρωπο και ανεξάρτητο από τη γλώσσα προγραμματισμού. Αποτελείται από ζεύγη, τύπου χαρακτηριστικό-τιμή, (key-value) και χρησιμοποιείται κυρίως για τη μετάδοση δεδομένων μεταξύ του εξυπηρετητή και των διαδικτυακών εφαρμογών, ως εναλλακτική λύση της XML.

```
search_results_2.jsp
100     int i=0;
101     while (rs.next()) {
102
103         //Creating a JSONObject object
104         obj = new JSONObject();
105
106         //Inserting key-value pairs into the json object
107         obj.put("lat",rs.getString("latitude"));
108         obj.put("lng",rs.getString("longitude"));
109         obj.put("num",rs.getString("num_studies"));
110         obj.put("city",rs.getString("city"));
111
112         i++;
113
114         marker.put(obj);
115
116     }
117
118     <tr>
119         <td><%=rs.getString("city")%></td>
120         <td><%=rs.getString("num_studies")%></td>
121     </tr>
122
123     <%
124     }
125     %>
126
127 </table>
128
129     <%
130     String jsonText = marker.toString();
131     //java.lang.String markers =jsonText;
132     // out.println(jsonText);
133
134     // out.println(application.getRealPath(""));
135
136     // Δημιουργία json αρχείου
137     FileWriter filewrt = new FileWriter(application.getRealPath("/marker.json"));
138     BufferedWriter fileout = new BufferedWriter(filewrt);
139     fileout.write(jsonText);
140     fileout.close();
141
142     // String fileName = application.getRealPath("/marker.json");
143     // System.out.println(fileName+" records");
144
145     rs.close();
146
147     }
148     %>
```

# ΠΑΡΟΥΣΙΑΣΗ ΚΩΔΙΚΑ

## Παρουσίαση αποτελεσμάτων σε χάρτη – **map.jsp**

Όπως φαίνεται στον κώδικα **map.jsp**, για την παρουσίαση των αποτελεσμάτων σε χάρτη χρησιμοποιήθηκαν οι βιβλιοθήκες JavaScript APIs της Google για maps (γραμμή 30), όπως επίσης και jQuery CDN\* από τη Google (γραμμή 33). Για το map container χρειάζεται η δημιουργία ενός HTML element, id="map", στο οποίο καθορίζεται και το μέγεθός του width και height (γραμμής 35).

Το JavaScript, **map.js**, για την εμφάνιση του χάρτη καλείται στη γραμμή 38. Το αναφερόμενο JavaScript θα μπορούσε να έχει ενσωματωθεί στον κώδικα map.jsp, αλλά για εκπαιδευτικές ανάγκες υλοποιήθηκε ξεχωριστά.

```
map.jsp x
1  <%@ page language="java" contentType="text/html; charset=UTF-8"
2  pageEncoding="UTF-8"%>
3  <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
4  <%@ page import="java.sql.*"%>
5
6  <!-- ΕΜΦΑΝΙΣΗ ΤΩΝ ΠΟΛΕΩΝ ΣΤΙΣ ΟΠΟΙΕΣ ΕΛΑΒΑΝ ΧΩΡΑ ΚΛΙΝΙΚΕΣ ΜΕΛΕΤΕΣ ΠΑΝΩ ΣΕ ΧΑΡΤΗ ΚΑΙ ΤΟΥ ΠΛΗΘΟΥΣ ΜΕΛΕΤΩΝ ΩΣ ΠΛΗΡΟΦΟΡΙΑ -->
7
8  <html>
9
10
11 <head>
12     <title lang='gr'>Project CTG0V-03</title>
13     <link rel='icon' href='prometheus.png' type='image/png'/>
14     <meta name='viewport' content='width=device-width, initial-scale=1.0'></meta>
15     <link rel='stylesheet' href='common.css'></link>
16 </head>
17
18 <body>
19
20 <center><a href="home.html"></a>
21
22 <div class="navbar" id='navbar'>
23     <a href="home.html">Αρχική Σελίδα</a>
24     <a href="about.html">Σχετικά Με</a>
25 </div>
26
27 <h1> ΠΟΛΕΙΣ ΣΤΙΣ ΟΠΟΙΕΣ ΕΛΑΒΑΝ ΧΩΡΑ ΚΛΙΝΙΚΕΣ ΜΕΛΕΤΕΣ </h1>
28
29 <!-- The Google Maps JavaScript API is loaded -->
30 <script src="http://maps.google.com/maps/api/js?sensor=false" type="text/javascript"></script>
31
32 <!-- jQuery CDN (Content Delivery Network) from Google -->
33 <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.0/jquery.min.js"></script>
34 <!-- Εμφάνιση χάρτη που περιέχει και αναφορά μέγεθους χάρτη -->
35 <div id="map" style="width: 1000px; height: 1000px;"></div>
36
37
38 <script type="text/javascript" src="map.js"></script>
39
40
41 </center>
42
43 </body>
```

\*Content Delivery Network

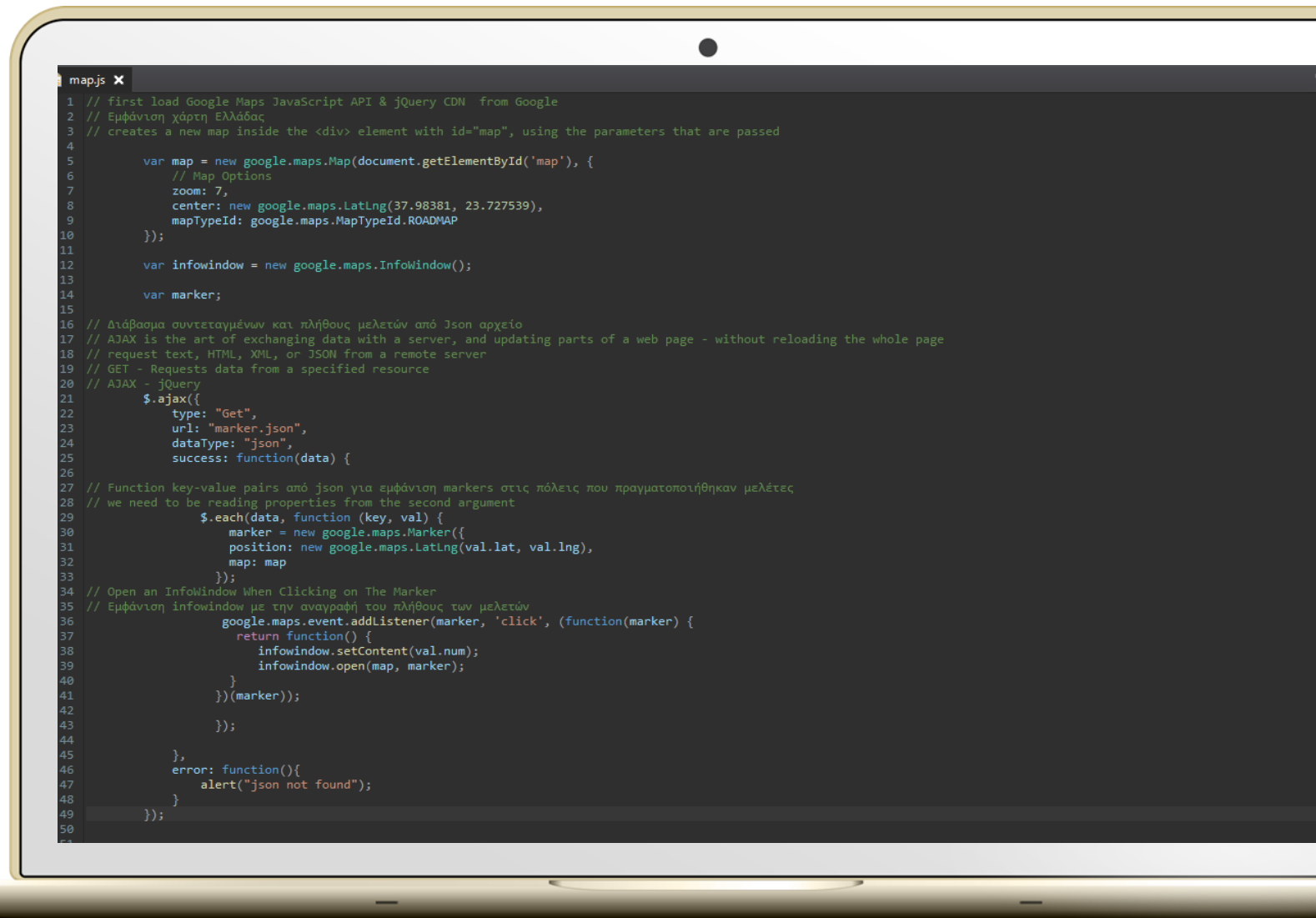
# ΠΑΡΟΥΣΙΑΣΗ ΚΩΔΙΚΑ

## Παρουσίαση αποτελεσμάτων σε χάρτη (I) – map.js

Αρχικά, στο JavaScript **map.js** με τις εντολές των γραμμών 5-10 καθορίζονται τα options για την εμφάνιση του χάρτη της Ελλάδας.

Στη συνέχεια, για να μπορέσει ο client να χρησιμοποιήσει τις συντεταγμένες των ελληνικών πόλεων και τον αριθμό των κλινικών μελετών (για τις ανάγκες εμφάνισης των markers) τα οποία βρίσκονται στο **JSON** αρχείο, χρησιμοποιείται η τεχνική **AJAX\*** με **jQuery**.

Η **AJAX** είναι μια ομάδα αλληλένδετων τεχνικών ανάπτυξης ιστοσελίδων, που χρησιμοποιούνται στην πλευρά του client, για να δημιουργήσουν ασύγχρονες διαδικτυακές εφαρμογές. Με την AJAX, οι εφαρμογές ιστού μπορούν να στείλουν και να ανακτήσουν δεδομένα από έναν server ασύγχρονα, δηλαδή στο παρασκήνιο, χωρίς να παρεμβαίνουν στην οθόνη και τη συμπεριφορά της υπάρχουσας σελίδας.



\*Asynchronous JavaScript and XML



# ΠΑΡΟΥΣΙΑΣΗ ΚΩΔΙΚΑ

## Παρουσίαση αποτελεσμάτων σε χάρτη (II) – [map.js](#)

Η **jQuery** μας παρέχει μια πλούσια γκάμα από συναρτήσεις για τον προγραμματισμό και την χρήση της AJAX. Με την jQuery μπορούμε να κάνουμε αιτήσεις TXT, HTML, XML ή JSON σε έναν απομακρυσμένο Server χρησιμοποιώντας είτε το Get είτε το Post. Έτσι μπορούμε να φορτώσουμε απομακρυσμένα δεδομένα απευθείας σε κάποιο στοιχείο της ιστοσελίδας μας.

Στον κώδικα με την μέθοδο “**\$.ajax**” γίνεται request, **type=“Get”**, για το JSON αρχείο, **datatype=“json”**, από συγκεκριμένη URL, **url:“marker.json”**. Στην παρούσα εφαρμογή το URL είναι το localhost. Στην περίπτωση που το request είναι **success**, επιστρέφονται από τον server τα data.

```
map.js x
1 // first load Google Maps JavaScript API & jQuery CDN from Google
2 // Εμφάνιση χάρτη Ελλάδας
3 // creates a new map inside the <div> element with id="map", using the parameters that are passed
4
5 var map = new google.maps.Map(document.getElementById('map'), {
6     // Map Options
7     zoom: 7,
8     center: new google.maps.LatLng(37.98381, 23.727539),
9     mapTypeId: google.maps.MapTypeId.ROADMAP
10 });
11
12 var infowindow = new google.maps.InfoWindow();
13
14 var marker;
15
16 // Διάβασμα συντεταγμένων και πλήθους μελετών από Json αρχείο
17 // AJAX is the art of exchanging data with a server, and updating parts of a web page - without reloading the whole page
18 // request text, HTML, XML, or JSON from a remote server
19 // GET - Requests data from a specified resource
20 // AJAX - jQuery
21 $.ajax({
22     type: "Get",
23     url: "marker.json",
24     dataType: "json",
25     success: function(data) {
26
27         // Function key-value pairs από json για εμφάνιση markers στις πόλεις που πραγματοποιήθηκαν μελέτες
28         // we need to be reading properties from the second argument
29         $.each(data, function (key, val) {
30             marker = new google.maps.Marker({
31                 position: new google.maps.LatLng(val.lat, val.lng),
32                 map: map
33             });
34
35             // Open an InfoWindow When Clicking on The Marker
36             // Εμφάνιση infowindow με την αναγραφή του πλήθους των μελετών
37             google.maps.event.addListener(marker, 'click', (function(marker) {
38                 return function() {
39                     infowindow.setContent(val.num);
40                     infowindow.open(map, marker);
41                 }
42             })(marker));
43
44         });
45
46     },
47     error: function(){
48         alert("json not found");
49     }
50 });
```

# ΠΑΡΟΥΣΙΑΣΗ ΚΩΔΙΚΑ

## Παρουσίαση αποτελεσμάτων σε χάρτη (III) – [map.js](#)

Έπειτα με τη μέθοδο “\$.each” εκτελείται συνάρτηση που επιστρέφει διαδοχικά τις τιμές για το γεωγραφικό πλάτος και μήκος και για το πλήθος των μελετών. Η εμφάνιση των markers επιτυγχάνεται με τη δημιουργία ενός στιγμιότυπου της JavaScript κλάσης με όνομα Marker. Για την εμφάνιση των marker πάνω στο χάρτη είναι απαραίτητες οι τιμές των συντεταγμένων (val.lat, val.lng από το json αρχείο).

Από την άλλη, το πλήθος των μελετών (val.num από το json αρχείο) εμφανίζεται στο info-window, το οποίο ανοίγει με το event “κλικ” του χρήστη πάνω στον κάθε marker.

```
map.js x
1 // first load Google Maps JavaScript API & jQuery CDN from Google
2 // Εμφάνιση χάρτη Ελλάδας
3 // creates a new map inside the <div> element with id="map", using the parameters that are passed
4
5 var map = new google.maps.Map(document.getElementById('map'), {
6     // Map Options
7     zoom: 7,
8     center: new google.maps.LatLng(37.98381, 23.727539),
9     mapTypeId: google.maps.MapTypeId.ROADMAP
10 });
11
12 var infowindow = new google.maps.InfoWindow();
13
14 var marker;
15
16 // Διάβασμα συντεταγμένων και πλήθους μελετών από json αρχείο
17 // AJAX is the art of exchanging data with a server, and updating parts of a web page - without reloading the whole page
18 // request text, HTML, XML, or JSON from a remote server
19 // GET - Requests data from a specified resource
20 // AJAX - jQuery
21 $.ajax({
22     type: "Get",
23     url: "marker.json",
24     dataType: "json",
25     success: function(data) {
26
27         // Function key-value pairs από json για εμφάνιση markers στις πόλεις που πραγματοποιήθηκαν μελέτες
28         // we need to be reading properties from the second argument
29         $.each(data, function(key, val) {
30             marker = new google.maps.Marker({
31                 position: new google.maps.LatLng(val.lat, val.lng),
32                 map: map
33             });
34
35             // Open an InfoWindow When Clicking on The Marker
36             // Εμφάνιση infowindow με την αναγραφή του πλήθους των μελετών
37             google.maps.event.addListener(marker, 'click', (function(marker) {
38                 return function() {
39                     infowindow.setContent(val.num);
40                     infowindow.open(map, marker);
41                 }
42             })(marker));
43
44         });
45
46     },
47     error: function(){
48         alert("json not found");
49     }
50 });
```

Λεπτομερής ανάλυση του κώδικα για όλο το project παρουσιάζεται στο βίντεο.



## CTG0V-03 Project

Ευχαριστώ πολύ για την υπομονή  
και το χρόνο που μου διαθέσατε.

👤 Ιωάννης Παπίας

✉ [el16058@mail.ntua.gr](mailto:el16058@mail.ntua.gr)

🔗 <https://github.com/loannisPapias/CTG0V-03Project>

🔗 [https://www.youtube.com/watch?v=ySPH31K77Y  
&feature=youtu.be](https://www.youtube.com/watch?v=ySPH31K77Y&feature=youtu.be)

