

ΠΡΟΧΩΡΗΜΕΝΑ ΘΕΜΑΤΑ
ΤΕΧΝΟΛΟΓΙΑΣ ΚΑΙ ΕΦΑΡΜΟΓΩΝ
ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ

ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΕΡΓΑΣΙΑ ΓΙΑ
ΤΟ ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2021-2022

ΟΜΑΔΑ

ΕΥΑΓΓΕΛΟΣ ΗΛΙΑΔΗΣ, ΑΜ 3117

ΒΑΣΙΛΗΣ ΝΙΚΑΣ, ΑΜ 3143

ΤΕΛΙΚΗ ΑΝΑΦΟΡΑ

ΜΑΪΟΣ 2022

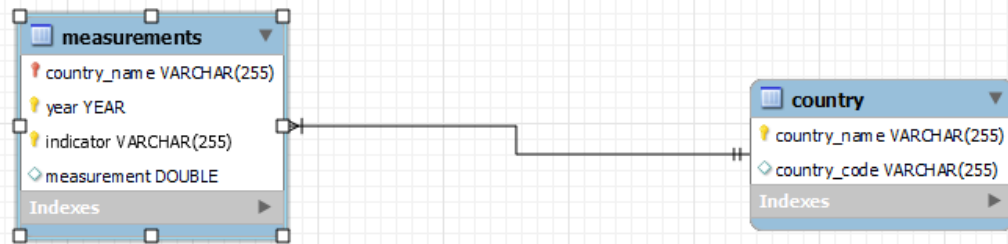
ΙΣΤΟΡΙΚΟ ΠΡΟΗΓΟΥΜΕΝΩΝ ΕΚΔΟΣΕΩΝ

Ημερομηνία	Έκδοση	Περιγραφή	Συγγραφέας
2022/03/08	1.0	Ολοκλήρωση της ΦΑΣΗΣ 1 <ul style="list-style-type: none">• Data extraction from csv files• DB setup/ connection establishment• Load data to DB	
2022/04/07	2.0	Ολοκλήρωση της ΦΑΣΗΣ 2 <ul style="list-style-type: none">• Web app setup• Main Controller• Different html templates	
2022/05/24	2.1	<ul style="list-style-type: none">• Σχεδίαση διαδραστικού περιβάλλοντος• Φόρμα αναζήτησης• Φόρτωμα αρχικού υποδειγματικού διαγράμματος	

1 ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ

Στην παρούσα ενότητα περιγράφονται τα σχήματα της βάσης (ή βάσεων, αν είναι παραπάνω από μία) δεδομένων που χρησιμοποιούνται στο project.

1.1 ΣΧΕΣΙΑΚΟ ΣΧΗΜΑ ΣΕ ΛΟΓΙΚΟ ΕΠΙΠΕΔΟ



Σχήμα 1.1 Σχεσιακό σχήμα της βάσης δεδομένων του συστήματος

```
CREATE TABLE country(  
country_name VARCHAR(255) ,  
country_code VARCHAR(255),  
PRIMARY KEY(country_name))ENGINE = INNODB
```

```
CREATE TABLE measurements(  
country_name VARCHAR(255),  
year YEAR(4),  
indicator VARCHAR(255),  
measurement DOUBLE DEFAULT NULL  
,FOREIGN KEY(country_name) REFERENCES country(country_name),  
PRIMARY KEY(country_name,year,indicator))ENGINE = INNODB
```

Το κομμάτι της MySQL, που είναι υπεύθυνο για το table creation, είναι μέρος του python script "pyScript2_0.py".

1.2 ΣΧΕΣΙΑΚΟ ΣΧΗΜΑ ΣΕ ΦΥΣΙΚΟ ΕΠΙΠΕΔΟ

1.2.1 ΡΥΘΜΙΣΗ ΤΩΝ ΠΑΡΑΜΕΤΡΩΝ ΤΟΥ DBMS

- Αν και το storage engine από ότι διαπιστώσαμε είναι by default set ως INNODB. Κατά το table creation βάλαμε την παράμετρο ENGINE = INNODB.
- Θέσαμε το μέγεθος του buffer pool ίσο με 12GB ($\approx 80\%$ των συνολικών 16GB) στους προσωπικούς μας υπολογιστές, σύμφωνα με τις οδηγίες του 1^{ου} σετ ασκήσεων. Το μέγεθος του buffer pool πρέπει να είναι πολλαπλάσιο των 2GB.
- Επίσης θέσαμε το μέγεθος του logfile ίσο με $\frac{1}{4} * \text{buffer pool size} = \frac{1}{4} * 12 = 3\text{GB}$. Αν και γνωρίζουμε ότι είναι αρκετό το μέγεθος του να είναι ίσο με $\text{db_size} + 10/100 * \text{db_size}$. Μιας και το μέγεθος της βάσης μας βγήκε της τάξης των μερικών δεκάδων MB και το μέγεθος του logfile θα ήταν αρκετό να είναι της ίδιας τάξης.

1.2.2 ΡΥΘΜΙΣΗ ΤΟΥ ΦΥΣΙΚΟΥ ΣΧΗΜΑΤΟΣ ΤΗΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

1.2.3 ΡΥΘΜΙΣΗ ΑΣΦΑΛΕΙΑΣ

2 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΛΟΓΙΣΜΙΚΟΥ

2.1 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΚΑΙ ΔΟΜΗ ETL

1. Εκτέλεση του python script “pyStript2_0.py”.

1.1. Το πρώτο βήμα είναι η εκτέλεση του “pyStript2_0.py”. Ωστόσο πριν από αυτό πρέπει συμπληρωθεί το path του φακέλου που περιέχει τα csv αρχεία.

```
#FILE PATHS PART

for root,dirs,files in os.walk(''): #Folder path to where the csv files are stored at. Should be changed accordingly.
#It is HIGHLY suggested that all csv files are stored in the same folder so that none is skipped.
    for file in files:
        filename, extension = os.path.splitext(file)
        if extension == '.csv':
            paths.append('' + filename + extension)
```

(Σε περίπτωση πολλαπλών εκτελέσεων του script, πρέπει να διαγράφεται το αρχείο “ final_csv_file.csv”. Γιατί βρίσκεται στον ίδιο φάκελο με αυτόν από τα οποία λαμβάνονται τα αρχεία για data extraction. Επομένως σε περίπτωση πολλαπλών εκτελέσεων του script, θα υπάρχει πρόβλημα με το PRIMARY KEY των χωρών).

1.2. By default χρησιμοποιούμε τις ακόλουθες παραμέτρους για σύνδεση στο server της MySQL. Πρέπει να τροποποιηθούν ανάλογα τις προσωπικές ρυθμίσεις του καθενός.

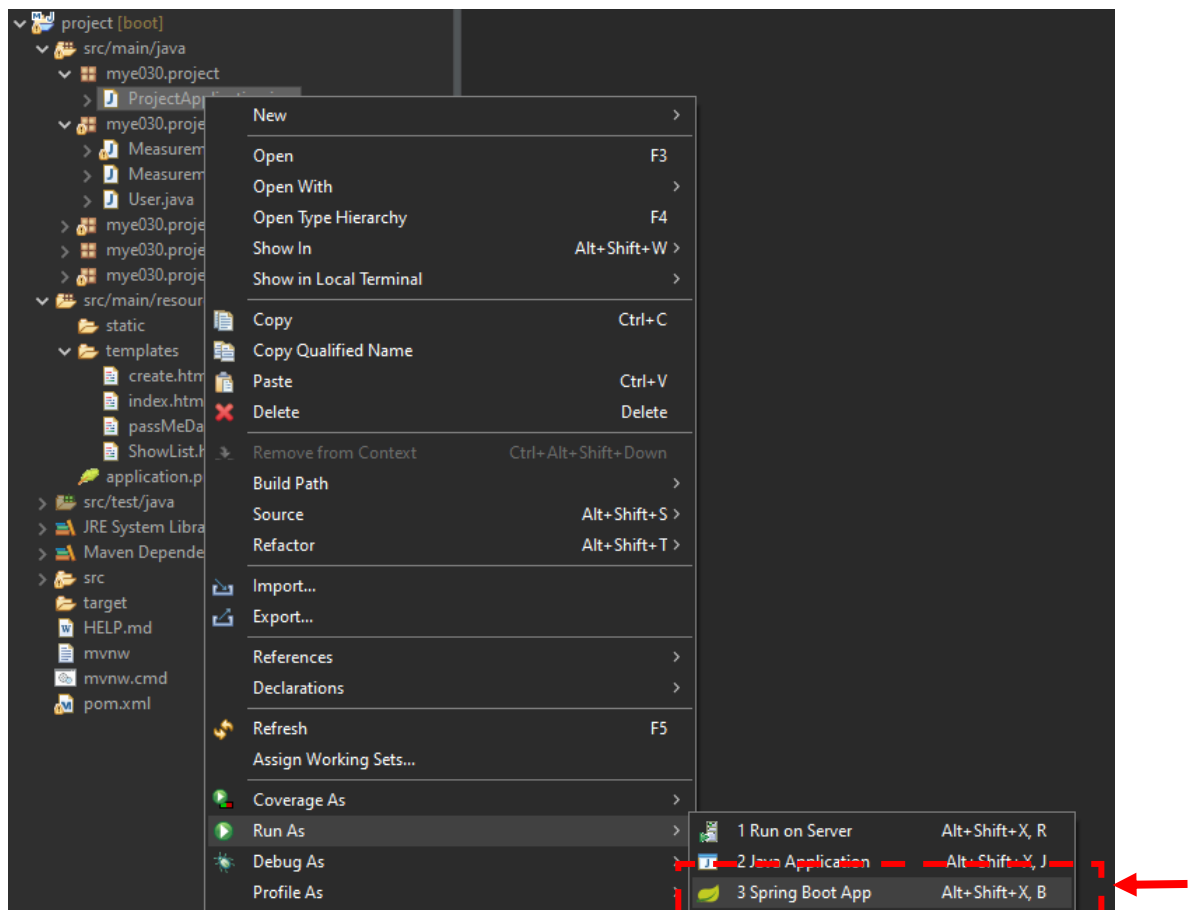
```
#CONNECTION PART

try: #trying to connect to MySQL.In case something goes wrong, exception will catch it.
    mydb = mysql.connector.connect(
        host="localhost",
        user="root",
        password="root",
        allow_local_infile=True
    )
```

- (Optional) Άνοιγμα του τελικού csv αρχείου και της βάσεις δεδομένων μέσω του MySQL Workbench, με σκοπό την εξακρίβωση της σωστής φόρτωσης των δεδομένων.
- Project/src/main/resources/application.properties
By default χρησιμοποιούμε τις ακόλουθες παραμέτρους για σύνδεση στο server της MySQL. Πρέπει να τροποποιηθούν ανάλογα τις προσωπικές ρυθμίσεις του καθενός.

```
1 spring.datasource.url = jdbc:mysql://localhost:3306/projectdb?useSSL=false&serverTimezone=UTC&useLegacyDateCode=false
2 spring.datasource.username = root
3 spring.datasource.password = root
4
```

- Εκτέλεση της main ως “Spring boot app”.



5. Μετάβαση στο url <http://localhost:8080/>.



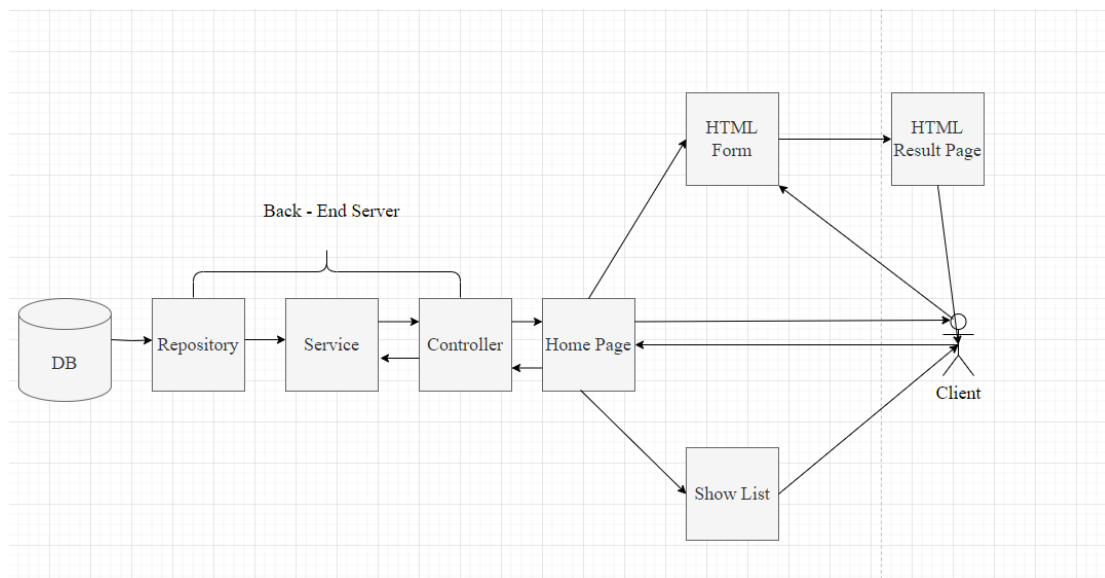
6. Επιλογή ενός εκ των δύο.

- Show list: Μετάβαση σε σελίδα που δείχνει όλα τα στοιχεία της βάσης
- Bar Chart: Μετάβαση στη φόρμα για συμπλήρωση των επιθυμητών αποτελεσμάτων



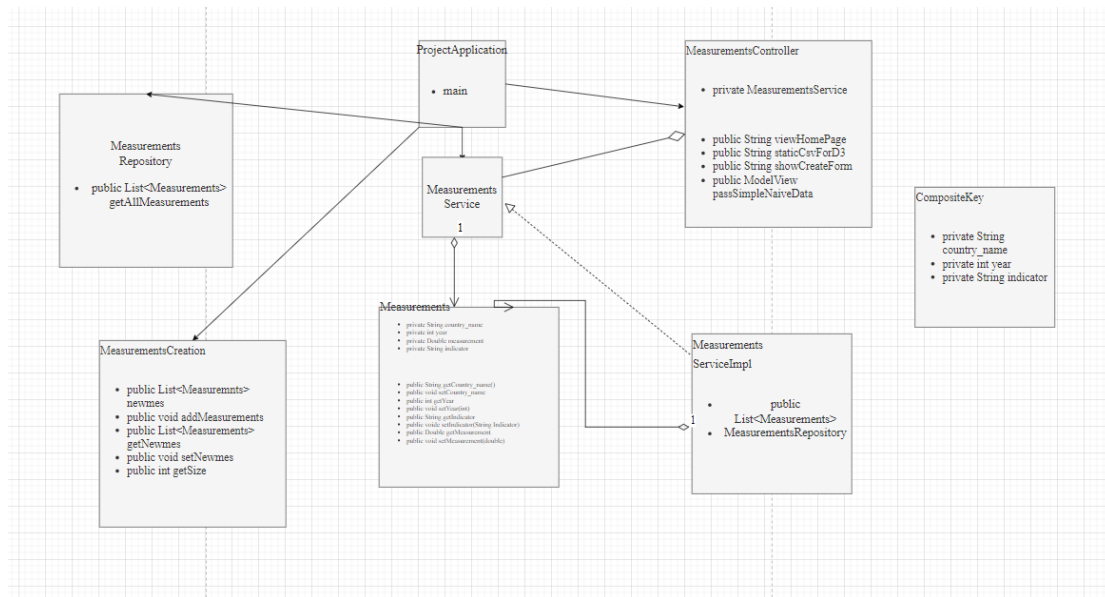
7. Show bar chart: Μετάβαση στη σελίδα προβολής του bar chart. (Ωστόσο η σελίδα περιέχει ένα διάγραμμα υπόδειγμα. Δεν ανταποκρίνεται στα δεδομένα που ζητάει ο χρήστης καθώς δεν μπορέσαμε να βρούμε τρόπο να παίρνουμε τα δεδομένα που δίνει ο χρήστης στη φόρμα)

2.2 ΔΙΑΓΡΑΜΜΑΤΑ ΠΑΚΕΤΩΝ / ΥΠΟΣΥΣΤΗΜΑΤΩΝ ΚΕΝΤΡΙΚΗΣ ΕΦΑΡΜΟΓΗΣ



2.3 ΔΙΑΓΡΑΜΜΑ(ΤΑ) ΚΛΑΣΕΩΝ ΚΕΝΤΡΙΚΗΣ ΕΦΑΡΜΟΓΗΣ

Διάγραμμα κλάσεων



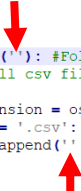
3 ΥΠΟΔΕΙΓΜΑΤΑ ΕΡΩΤΗΣΕΩΝ ΚΑΙ ΑΠΑΝΤΗΣΕΩΝ

- Αρχείο "pyScript2_0.py"
Συλλογή των paths των csv αρχείων
(Arrows indicate where the path to the csv files folder should be)

```

11 #FILE PATHS PART
12
13 for root,dirs,files in os.walk(''): #Folder path to where the csv files are stored at. Should be changed accordingly.
14     #It is HIGHLY suggested that all csv files are stored in the same folder so that none is skipped.
15     for file in files:
16         filename, extension = os.path.splitext(file)
17         if extension == '.csv':
18             paths.append('' + filename + extension)
19

```



Connection establishment με το server της MySQL

```

21 #CONNECTION PART
22
23 try: #trying to connect to MySQL.In case something goes wrong, exception will catch it.
24     mydb = mysql.connector.connect(
25         host="localhost",
26         user="root",
27         password="root",
28         allow_local_infile=True
29     )
30     if mydb.is_connected():
31         cursor = mydb.cursor()
32
33         try:
34             cursor.execute("DROP DATABASE " + dbName) #First we drop the DB, in case the DB already exists.
35         except Error as e:
36             print("Couldn't drop db. Db does not exist")
37         cursor.execute("CREATE DATABASE " + dbName) #DB creation.
38
39         print("Database is created")
40
41 except Error as e:
42     print("Encountered an error while trying to connect to MySQL :", e)
43
44 cursor.execute("USE " + dbName)
45 cursor.execute("SET GLOBAL sql_mode=''")
46 cursor.execute("SET GLOBAL local_infile=1")

```

Table creation

```

#Table creation. Setting InnoDB as storage engine.
cursor.execute("CREATE TABLE country(country_name VARCHAR(255),country_code VARCHAR(255),PRIMARY KEY(country_name))ENGINE = InnoDB")
cursor.execute("CREATE TABLE measurements(country_name VARCHAR(255),year YEAR(4),indicator VARCHAR(255),measurement DOUBLE DEFAULT NULL,FOREIGN KEY(country_name) REFERENCES country(country_name),PRIMARY KEY(country_name,year,indicator))ENGINE = InnoDB")

```


Load data from csv file to DB

(Final csv file path)



```
cursor.execute("LOAD DATA LOCAL INFILE '...' INTO TABLE measurements FIELDS  
TERMINATED BY ',' OPTIONALLY ENCLOSED BY '\"' LINES TERMINATED BY '\r\n'  
(country_name,year,indicator,@vmeasurement) SET measurement =  
NULLIF(@vmeasurement,'')")
```

```
mydb.commit()
```

- Java Application
 - Controller
 - Main page handler

```
44  
45 @GetMapping("/")  
46 public String viewHomePage(Model model) {  
47     model.addAttribute("listMeasurements", measurementsService.getAllMeasurements());  
48     //for(int i=0;i < measurementsService.getAllMeasurements().size();i++) {  
49         // System.out.println(measurementsService.getAllMeasurements().get(i).getIndicator());  
50     }  
51     return "index";  
52 }
```

Show List page handler

```

@RequestMapping("/ShowList")
public String staticCsvForD3(Model model) {
    List<Measurements> dokimh =new ArrayList<Measurements>(measurementsService.getAllMeasurements());
    List<Integer> xronies =new ArrayList<Integer>();
    List<Double> measur =new ArrayList<Double>();
    List<String> indicator =new ArrayList<String>();
    List<String> cname =new ArrayList<String>();
    //xronies.add(1);

    for(Measurements i: dokimh) {
        //System.out.println(i.getYear());
        //for (int j: xronies) {
        if(!xronies.contains(i.getYear())) {
            xronies.add(i.getYear());
            //System.out.println(i.getYear());
            //System.out.println(xronies.size());
        }
        if(!measur.contains(i.getMeasurement())) {
            if (i.getMeasurement() != null) {
                measur.add(i.getMeasurement());
                //System.out.println(i.getYear());
                //System.out.println(xronies.size());
            }
        }
        if(!cname.contains(i.getCountry_name())) {
            cname.add(i.getCountry_name());
            //System.out.println(i.getYear());
            //System.out.println(xronies.size());
        }
        if(!indicator.contains(i.getIndicator())) {
            indicator.add(i.getIndicator());
            //System.out.println(i.getYear());
            //System.out.println(xronies.size());
        }
    }
    model.addAttribute("xronies", xronies);
    model.addAttribute("cname", cname);
    model.addAttribute("indicators", indicator);
    //model.addAttribute("apotelesmata", dokimh);
    return "/ShowList";
}

```

Create page handler

```

@RequestMapping(value = "/create", method = {RequestMethod.GET, RequestMethod.POST})
public String showCreateForm(Model model, HttpServletRequest req) {
    //MeasurementsCreation selectForm = new MeasurementsCreation();
    //List<Measurements> dokimh =new ArrayList<Measurements>();

    selectForm.addMeasurement(new Measurements());

    model.addAttribute(new User());
    //model.addAttribute("form", selectForm);
    for (int i = 0; i < selectForm.getSize(); i++) {
        System.out.println(selectForm.getNewmes().get(0).getYear());
    }
    String n1 = req.getParameter("labelValue");
    //System.out.println(n1);
    //selectForm.getNewmes().get(0).setCountry_name(model.getAttribute(selectForm.getNewmes().get(0).getCountry_name()));
    return "/create";
}

```

Diagram page handler

```

@RequestMapping(value = "/passMeDataNaive03", method = {RequestMethod.GET, RequestMethod.POST})
public ModelAndView passSimpleNaiveData(@ModelAttribute User user, Model model, HttpServletRequest req, HttpServletResponse response) {

    List<Measurements> dokimh = new ArrayList<Measurements>(measurementsService.getAllMeasurements());
    List<Integer> xronies = new ArrayList<Integer>();
    List<Double> measur = new ArrayList<Double>();
    List<String> indicator = new ArrayList<String>();
    List<String> cname = new ArrayList<String>();
    //xronies.add(1);

    model.addAttribute("user", user);
    System.out.println(user.getName());
    /*MeasurementsService helperMeasurementService;

    measurementsService.saveAll(form.getNewmes());

    modell.addAttribute("books", helperMeasurementService.getAllMeasurements());*/

    //selectForm.getNewmes().get(0).setIndicator("aaaaa");
    //dokimh = measurementsService.saveAllMeasurements(selectForm.getNewmes());

    for(Measurements i: dokimh) {
        //System.out.println(i.getYear());
        //for (int j: xronies) {
        if(!xronies.contains(i.getYear())) {
            xronies.add(i.getYear());
            //System.out.println(i.getYear());
            //System.out.println(xronies.size());
        }
        if(!measur.contains(i.getMeasurement())) {
            if (i.getMeasurement() != null) {
                measur.add(i.getMeasurement());
                //System.out.println(i.getYear());
                //System.out.println(xronies.size());
            }
        }
    }

    if(!cname.contains(i.getCountry_name())) {
        cname.add(i.getCountry_name());
        //System.out.println(i.getYear());
        //System.out.println(xronies.size());
    }
    if(!indicator.contains(i.getIndicator())) {
        indicator.add(i.getIndicator());
        //System.out.println(i.getYear());
        //System.out.println(xronies.size());
    }
}

/*for (int j: xronies) {
    System.out.println(j);
}
for (double j: measur) {
    System.out.println(j);
}*/
//showCreateForm(modell);
Map<Integer, Double> values = new HashMap<Integer, Double>();
values.put(2020, 135.0);values.put(2021, 125.0);values.put(2022, 45.0);

/* Sort the map: (a) make a sorted AL; (b) put them in a string in order
 * The string also has a "header" with the names of the attributes in each pair.
 * */
List<Integer> sortedKeys=new ArrayList<Integer>(values.keySet());
Collections.sort(sortedKeys);

String s ="key, value\n";
for(Integer i: sortedKeys) {
    Double v = values.get(i);
    s = s + i + ", " + v + "\n";
}
}

```

```
String constructedText = "This is some text that accompanies the data sent to the page. "
    + "You can generate such a text to describe the query characteristics.";
String pageTitle = "Bar Chart";

/* Convert the string to json
 * */
//JSONObject jo = new JSONObject(values);
//model.addAttribute("dataMap", jo);
//System.out.println ("\\n\\n" + jo.toString() + "\\n\\n");
JSONArray result = CDL.toJSONArray(s);
model.addAttribute("dataMap", result);
model.addAttribute("description", constructedText);
model.addAttribute("title", pageTitle);

response.setContentType("text/html");
String[] values1 = req.getParameterValues("labelValue");
String n1 = req.getParameter("labelValue");
//List<String> helper= new ArrayList<String>(Arrays.asList(values));
for (int i = 0; i < selectForm.getSize(); i++) {
    //System.out.println(n1);
    //System.out.println(values1[i]);
}
return new ModelAndView("passMeDataNaiveD3");
}
}
```

4 ΛΟΙΠΑ ΣΧΟΛΙΑ

Επιχειρήσαμε πολλές διαφορετικές υλοποιήσεις εξαγωγής των στοιχείων που δηλώνει ο χρήστης στην φόρμα και διαχείρησής τους στο java servlet. Ωστόσο όλες οι προσπάθειες ήταν ανεπιτυχείς. Επομένως χωρίς το feedback του χρήστη για το ποια στοιχεία ήθελε στα διαγράμματα δεν μπορέσαμε να συνεχίσουμε παρακάτω στην υλοποίηση των διαγραμματικών στοιχείων.