

**ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ**

**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ**

**ΣΧΕΔΙΟ ΠΑΡΟΥΣΙΑΣΗΣ  
ΑΤΟΜΙΚΗΣ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ**

Μάιος 2024

Ατομική Διπλωματική Εργασία

**ΕΡΓΑΛΕΙΟ ΣΥΛΛΟΓΗΣ ΔΕΔΟΜΕΝΩΝ ΑΠΟ ΤΟ MAVEN**

Αδάμος Κυριάκου

**ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ**



**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ**

**Μάιος 2024**

**ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ**

**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ**

**ΕΡΓΑΛΕΙΟ ΣΥΛΛΟΓΗΣ ΔΕΔΟΜΕΝΩΝ ΑΠΟ ΤΟ MAVEN**

**Αδάμος Κυριάκου**

Επιβλέπων Καθηγητής

Ελένη Κωνσταντίνου

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των  
απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του  
Πανεπιστημίου Κύπρου

Μάιος 2024

# Ευχαριστίες

Θα ήθελα να ευχαριστήσω την επιβλέπουσα καθηγήτρια μου, Κυρία Ελένη Κωνσταντίνου για όλη την υποστήριξη και την βοήθεια που είχα για την υλοποίηση της διπλωματικής μου εργασίας.

# Περίληψη

Η διπλωματική εργασία επικεντρώνεται στον τομέα της διαχείρισης εξαρτήσεων πακέτων στην ανάπτυξη λογισμικού, με έμφαση στη γλώσσα προγραμματισμού Java και στο οικοσύστημα λογισμικού Maven. Στο πλαίσιο αυτού του τομέα διεξήχθη έρευνα και υλοποιήθηκε ένα πρόγραμμα που περιλαμβάνει τη δημιουργία μιας βάσης δεδομένων για την αποθήκευση δεδομένων, καθώς και μιας ιστοσελίδας που επιτρέπει στους χρήστες να αναζητούν εύκολα την εξάρτηση ενός πακέτου κατά την αναζήτηση στο Maven Central Repository. Για να γίνει αυτό, απαιτήθηκε η μελέτη για τα API γενικότερα και στην συνέχεια η Maven Central Repository. Πάρθηκε η απόφαση να υλοποιηθεί μια διαδικτυακή σελίδα με κεντρική λειτουργία την Maven Central Repository που θα βοηθούσε τους χρήστες να βρίσκουν τις πληροφορίες για τα πακέτα της Maven και τις εξαρτήσεις μεταξύ τους. Πιο συγκεκριμένα, η σελίδα αυτή προσφέρει την αναζήτηση βάση διαφορετικών πεδίων ώστε να κάνει την πλοήγηση των χρηστών στη σελίδα πιο ενδιαφέρουσα. Τέλος, μέσα από κάποιες βελτιώσεις τις οποίες θα αναλύσουμε στη συνέχεια, διαφαίνεται ότι τα επόμενα χρόνια είναι δυνατόν για τους χρήστες της Maven να την χρησιμοποιούν ως επιπρόσθετο εργαλείο για ευκολότερη πλοήγηση στην Maven Central Repository αφού μέσω των πολλών πεδίων που παρέχονται οι χρήστες θα μπορούν να βρουν πακέτα εύκολα και γρήγορα. [Adducul, R. (2020)] [1].

# Περιεχόμενα

|                                                          |           |
|----------------------------------------------------------|-----------|
| <b>Κεφάλαιο 1 Εισαγωγή.....</b>                          | <b>1</b>  |
| 1.1 Γενική ιδέα Διπλωματικής                             | 1         |
| 1.2 Κίνητρα                                              | 2         |
| 1.3 Στόχοι Διπλωματικής                                  | 3         |
| <br><b>Κεφάλαιο 2 Βασικές Έννοιες.....</b>               | <b>4</b>  |
| 2.1 Τι είναι το Maven και ποιες οι Χρησιμότητες του      | 4         |
| 2.2 Τι είναι τα Pom αρχεία της Maven                     | 5         |
| 2.3 Τι είναι οι Εξαρτήσεις και οι Βιβλιοθήκες στην Maven | 6         |
| 2.4 Γιατί κάποιος χρήστης να χρησιμοποιήσει την Maven    | 8         |
| <br><b>Κεφάλαιο 3 Απαιτήσεις.....</b>                    | <b>9</b>  |
| 3.1 Απαιτήσεις συστήματος                                | 9         |
| 3.2 Λειτουργικές απαιτήσεις                              | 9         |
| 3.3 Μη λειτουργικές απαιτήσεις                           | 10        |
| <br><b>Κεφάλαιο 4 Σχεδίαση συστήματος.....</b>           | <b>12</b> |
| 4.1 Αρχιτεκτονική συστήματος                             | 13        |
| 4.1.1 Υψηλού Επιπέδου Αρχιτεκτονική                      | 13        |
| 4.1.2 Σχεδίαση Βάσης Δεδομένων                           | 15        |
| 4.2 Σχεδίαση συστήματος                                  | 16        |
| 4.3 Front-end Development                                | 17        |
| 4.3.1 HTML5                                              | 18        |
| 4.3.2 CSS                                                | 19        |
| 4.3.3 JavaScript                                         | 19        |
| 4.3.4 BOOTSTRAP                                          | 20        |
| 4.4 Back-end Development                                 | 21        |
| 4.4.1 MYSQL                                              | 21        |
| 4.4.2 PHP                                                | 22        |

|                                                   |            |
|---------------------------------------------------|------------|
| 4.4.3 JSON                                        | 24         |
| 4.4.4 JDBC                                        | 24         |
| 4.4.5 JAVA                                        | 25         |
| 4.4.5.1 Jsoup                                     | 25         |
| 4.4.5.2 Javax.xml                                 | 26         |
| 4.4.5.3 org.w3c.dom                               | 27         |
| 4.4.5.4 org.xml.sax                               | 27         |
| <b>Κεφάλαιο 5 Λειτουργίες Συστήματος.....</b>     | <b>29</b>  |
| 5.1 Περιγραφή Λειτουργιών                         | 29         |
| <b>Κεφάλαιο 6 Αξιολόγηση.....</b>                 | <b>38</b>  |
| 6.1 MVN Repository                                | 38         |
| 6.2 Σύγκριση                                      | 39         |
| 6.2.1 Πλεονεκτήματα Συστήματος σε σχέση με το MVN | 39         |
| 6.2.2 Μειονεκτήματα Συστήματος σε σχέση με το MVN | 39         |
| 6.3 Αξιολόγηση συστήματος                         | 40         |
| <b>Κεφάλαιο 7 Συμπεράσματα .....</b>              | <b>41</b>  |
| 7.1 Επίλογος                                      | 42         |
| 7.2 Μελλοντική Δουλειά                            | 43         |
| <b>Βιβλιογραφία .....</b>                         | <b>43</b>  |
| <b>Παράρτημα Α.....</b>                           | <b>A-1</b> |
| <b>Παράρτημα Β.....</b>                           | <b>B-1</b> |
| <b>Παράρτημα Γ.....</b>                           | <b>Γ-1</b> |

# Κεφάλαιο 1

## Εισαγωγή

---

|                              |   |
|------------------------------|---|
| 1.1 Γενική ιδέα Διπλωματικής | 1 |
| 1.2 Κίνητρα                  | 2 |
| 1.3 Στόχοι Διπλωματικής      | 3 |

---

### 1.1 Γενική ιδέα Διπλωματικής

Η εποχή μας χαρακτηρίζεται από την γρήγορη εξέλιξη της τεχνολογίας, με νέα εργαλεία και προσεγγίσεις που αναδύονται συνεχώς για να ανταποκριθούν στις ανάγκες της ψηφιακής εποχής. Ένα από τα πιο ενδιαφέροντα γεγονότα που προκύπτουν από αυτήν την εξέλιξη είναι τα οικοσυστήματα λογισμικού. Τα οικοσυστήματα λογισμικού εκπροσωπούν ένα πλήθος εφαρμογών, εργαλείων και διαδικασιών που συνεργάζονται για να παρέχουν λύσεις σε τεχνολογικά προβλήματα [2]. Τα οικοσυστήματα αναγνωρίζονται για τη δυναμική τους φύση και την ικανότητά τους να αναπτύσσονται και να προσαρμόζονται ανάλογα με τις ανάγκες του χρήστη. Με βάση αυτό το πλαίσιο, η παρούσα διπλωματική εργασία επικεντρώνεται στη χρήση ενός οικοσυστήματος λογισμικού που χρησιμοποιεί την τεχνολογία της Maven ως έναν package manager που διαχειρίζεται πακέτα και βιβλιοθήκες. Χρησιμοποιώντας το οικοσύστημα αυτό, διερευνούμε τη δυνατότητα ανάκτησης δεδομένων από αρχεία pom.xml και την ενσωμάτωσή τους σε μια βάση δεδομένων. Μέσω ενός διαδικτυακού περιβάλλοντος που χρησιμοποιεί το οικοσύστημα, παρέχουμε στον χρήστη τη δυνατότητα αναζήτησης και προβολής των δεδομένων με έναν ευχάριστο και αποδοτικό τρόπο [3]. Ο κύριος λόγος που δημιουργήσαμε αυτό το σύστημα είναι για να βοηθήσουμε τους χρήστες να βρίσκουν και να μελετούν πιο εύκολα τα δεδομένα που επιθυμούν. Παρατηρήσαμε ότι στο Maven Central Repository, το κεντρικό αποθετήριο της Maven, οι χρήστες δεν παίρνουν τα δεδομένα που χρειάζονται αποδοτικά. Αυτό συμβαίνει επειδή ο χρήστης πρέπει να



εξετάζει κάθε πακέτο ξεχωριστά για να βρει τις αναγκαίες πληροφορίες. Με την δημιουργία αυτού του συστήματος, ο χρήστης μπορεί να βελτιώσει τη μελέτη του, καθώς το σύστημά παρέχει όλες τις πληροφορίες για κάθε πακέτο συγκεντρωμένες ακριβώς κάτω από αυτό.

## **1.2 Κίνητρα**

Ο βασικός μας στόχος είναι να δημιουργήσουμε ένα σύστημα που θα βελτιώσει την εμπειρία των χρηστών της Maven κατά τη μελέτη τους. Θέλουμε να διασφαλίσουμε ότι οι χρήστες θα έχουν μια ευχάριστη και αποτελεσματική εμπειρία κατά την περιήγησή τους στο σύστημα και την αναζήτηση πληροφοριών. Με αυτόν τον τρόπο, στοχεύουμε στη δημιουργία ενός περιβάλλοντος που θα καλύπτει τις ανάγκες τους και θα ενθαρρύνει την ενεργό συμμετοχή τους στην κοινότητα της Maven. Η ενημέρωση των χρηστών της Maven σχετικά με τη λειτουργία και τη σημασία των οικοσυστημάτων διευκολύνει τους χρήστες να κατανοήσουν και να αξιοποιήσουν σωστά τα εργαλεία και τις δυνατότητες που προσφέρονται. Μέσα από την δημιουργία αυτού του συστήματος, θέλουμε να βοηθήσουμε τους χρήστες της Maven Central Repository να ανακαλύψουν τις πληροφορίες ενός συγκεκριμένου πακέτου εύκολα και γρήγορα, χωρίς καθυστερήσεις καθώς και να αποθηκεύσουν τα δεδομένα τους σε διάφορες μορφές όπως CSV, XML, JSON [4].

Στο Maven Central Repository, ένα από τα βασικά προβλήματα είναι η δυσκολία που αντιμετωπίζουν οι χρήστες κατά την αναζήτηση πακέτων [5]. Καθώς το αποθετήριο είναι τεράστιο και περιέχει μεγάλο αριθμό πακέτων, οι χρήστες συχνά χρειάζεται να ξοδέψουν πολύ χρόνο για να εντοπίσουν το συγκεκριμένο πακέτο που χρειάζονται. Επιπλέον, η διαδικασία αναζήτησης μπορεί να είναι ακόμα πιο δύσκολη όταν οι χρήστες δεν γνωρίζουν ακριβώς το όνομα του πακέτου ή του συγγραφέα που ψάχνουν [6]. Επίσης, η πληροφορία για τα πακέτα και τις εξαρτήσεις τους ενδέχεται να μην είναι πάντα πλήρης ή κατανοητή, κάτι που μπορεί να δυσκολέψει τους χρήστες στην αξιολόγηση της καταλληλότητας ενός πακέτου.

### 1.3 Στόχοι Διπλωματικής

Για να μπορέσουμε να μετατρέψουμε τα κίνητρα που είχαμε σε μια υλοποίηση που να μπορέσει να βοηθήσει τους χρήστες που χρησιμοποιούν την Maven Central Repository θα πρέπει να ακολουθήσουμε κάποια συγκεκριμένα βήματα. Αρχικά θα γίνει μελέτη σε άρθρα για τα API γενικότερα και τις χρησιμότητες τους [Yang, T., & Cui, L. (2021)] [8]. Στην συνέχεια, θα γίνει μελέτη για τα πακέτα της Maven Central Repository μέσα από την ιστοσελίδα ώστε να ανακαλύψουμε τι περιλαμβάνεται σε κάθε πακέτο και πώς λειτουργεί. Μέσα από την έρευνα, θα καθοριστούν οι απαιτήσεις του συστήματος οι οποίες θα καταγράφουν και θα εξαχθούν οι βασικές λειτουργίες του συστήματος. Στο επόμενο στάδιο, θα προχωρήσουμε στην ανάπτυξη της εφαρμογής, χρησιμοποιώντας τη γλώσσα προγραμματισμού Java. Η εφαρμογή θα περιλαμβάνει τη λειτουργικότητα που επιτρέπει την ανάκτηση πληροφοριών από το Maven Central Repository και την οργάνωσή τους σε μια βάση δεδομένων. Επιπλέον, θα προχωρήσουμε στη δημιουργία μιας ιστοσελίδας η οποία θα παρέχει στους χρήστες πρόσβαση στο Maven Central Repository με έναν ευχάριστο και εύχρηστο τρόπο, επιτρέποντάς τους να αναζητούν πληροφορίες και να επικοινωνούν με τα πακέτα που τους ενδιαφέρουν με άνεση και αποτελεσματικότητα. Μέσω των δοκιμών που θα πραγματοποιηθούν, θα επιβεβαιώσουμε την ορθή λειτουργία του συστήματος. Η ύπαρξη πληθώρας δειγμάτων πακέτων θα επιτρέψει την εκτεταμένη δοκιμή της λειτουργικότητας του συστήματος και την επιβεβαίωση της απόδοσής του. Επίσης, θα εξεταστούν πιθανές βελτιώσεις, που μπορούν να γίνουν έτσι ώστε η εφαρμογή να είναι ακόμη πιο αποτελεσματική στο μέλλον.

## Κεφάλαιο 2

### Βασικές Έννοιες

---

|                                                          |   |
|----------------------------------------------------------|---|
| 2.1 Τι είναι το Maven και ποιές οι χρησιμότητες του      | 4 |
| 2.2 Τι είναι τα Pom αρχεία της Maven                     | 5 |
| 2.3 Τι είναι οι Εξαρτήσεις και οι Βιβλιοθήκες στην Maven | 6 |
| 2.4 Γιατί κάποιος χρήστης να χρησιμοποιήσει την Maven    | 8 |

---

#### 2.1 Τι είναι το Maven και ποιες οι χρησιμότητες του

Ένα κύριο μέρος της ανάπτυξης λογισμικού στην πλατφόρμα Java είναι η χρήση του Apache Maven. Το Maven είναι ένα ισχυρό εργαλείο λογισμικού και εξαρτήσεων καθώς η ονομασία του προέρχεται από τη λέξη “μαβέρα” στα αρχαία ελληνικά, που σημαίνει “γνώστης” ή “σοφός”. Μέσω του Maven, οι προγραμματιστές μπορούν να διαχειριστούν αυτόματα τις εξαρτήσεις του έργου τους, να δηλώσουν και να εκτελέσουν διαδικασίες κατασκευής, και να διαχειριστούν τις εκδόσεις του λογισμικού τους [9]. Επίσης, εγγυάται την αποτελεσματική διαχείριση των εξαρτήσεων του λογισμικού, επιτρέποντας την αυτόματη λήψη και εγκατάσταση των απαραίτητων πακέτων [10]. Με το Maven, οι προγραμματιστές μπορούν πλέον να διαχειρίζονται τα λογισμικά τους πιο αποτελεσματικά και σταθερά και να επιτύχουν υψηλότερη ποιότητα στο τελικό προϊόν τους.

Επιπλέον το Apache Maven προσφέρει πολλές χρησιμότητες που βοηθούν τις ομάδες ανάπτυξης να διαχειρίζονται και να αυτοματοποιούν τις διαδικασίες ανάπτυξης λογισμικού [11]. Ανάμεσα στις κύριες χρησιμότητες του Maven περιλαμβάνονται:

- Διαχείριση Εξαρτήσεων: Το Maven αναλαμβάνει τη διαχείριση των εξαρτήσεων του έργου, το οποίο επιτρέπει στους προγραμματιστές να καθορίζουν τις εξαρτήσεις στο αρχείο διαμόρφωσης (POM) [12] και να λαμβάνουν αυτόματα τις απαιτούμενες βιβλιοθήκες από αποθετήρια.
- Αυτοματοποίηση Κατασκευής: Η συγκέντρωση του κώδικα, η επαλήθευση δοκιμής μονάδας, και η δημιουργία εκτελέσιμων αρχείων είναι μερικές από τις λειτουργίες κατασκευής που ενδέχεται να προγραμματιστούν να εκτελούνται αυτόματα χρησιμοποιώντας το Maven [13].
- Διαχείριση Έργων: Το Maven προσφέρει διάφορα εργαλεία για τη διαχείριση πολυμορφικών έργων, επιτρέποντας τη διαχείριση πολλαπλών μονάδων και εκδόσεων [14].
- Ενσωμάτωση με Άλλα Εργαλεία: Το Maven διευκολύνει μια ομαλή και αποτελεσματική διαδικασία ανάπτυξης, ενσωματώνοντας άλλα εργαλεία όπως περιβάλλοντα υλοποίησης (CI/CD) και διακομιστές αυτοματοποιημένης διανομής (SDS) [15].

Με όλες αυτές τις χρησιμότητες, το Maven είναι ένα απαραίτητο εργαλείο για τις ομάδες ανάπτυξης λογισμικού, που συμβάλλει στην αύξηση της παραγωγικότητας και της ποιότητας του παραγόμενου κώδικα.

## 2.2 Τι είναι τα Pom αρχεία της Maven

Τα κύρια αρχεία διαμόρφωσης στο Maven για ένα έργο ονομάζονται Project Object Model (POM) [16]. Αυτά τα αρχεία καθορίζουν τη δομή του έργου και τις εξαρτήσεις του από άλλα προγράμματα, βιβλιοθήκες και εξωτερικές πηγές. Στο Maven, κάθε έργο πρέπει να περιέχει ένα αρχείο POM που ονομάζεται "pom.xml" και βρίσκεται στον κατάλογο του έργου. Τα αρχεία POM περιλαμβάνουν δεδομένα όπως στοιχεία έκδοσης, διαχείριση εξαρτήσεων, διαχείριση αποθετηρίων (repositories) και συμβάντα build (build lifecycle) [17]. Τα στοιχεία έκδοσης περιλαμβάνουν το όνομα του έργου, την έκδοση,

τον προγραμματιστή, την άδεια χρήσης και άλλα σχετικά δεδομένα. Η διαχείριση εξαρτήσεων καθορίζεται από ποιες βιβλιοθήκες ή έργα Java εξαρτάται το πρόγραμμα. Σκοπός της διαχείρισης αποθετηρίων είναι να προσδιορίζουν τις πηγές από τις οποίες το Maven θα λάβει εξαρτήσεις έργου [18]. Μετέπειτα, το build lifecycle ορίζει τα στάδια διαδικασίας της κατασκευής του προγράμματος, συμπεριλαμβανομένης της συλλογής κώδικα, της μεταγλώττιση, τη δοκιμή, της συσκευασία και της ανάπτυξης [19]. Τα αρχεία POM είναι απαραίτητα για τη λειτουργικότητα του Maven, καθώς περιλαμβάνουν τις οδηγίες και τις πληροφορίες που απαιτούνται για τη επίβλεψη και την αυτοματοποίηση των διαδικασιών ανάπτυξης έργου. Στο σύστημα που δημιουργήσαμε αυτά τα αρχεία τα πέραμε με το url της Maven Central Repository, βγάσαμε όλη την πληροφορία που είχαν μέσα τους, τα επεξεργαζόμασταν, και τα βάζαμε στην βάση δεδομένων μας ώστε να μπορεί οποιαδήποτε στιγμή ο χρήστης όταν τα ζητήσει να του τα παρέχουμε.

### **2.3 Τι είναι οι εξαρτήσεις και οι βιβλιοθήκες στην Maven**

Ένα σημαντικό κομμάτι στη διαδικασία ανάπτυξης λογισμικού είναι η διαχείριση των εξαρτήσεων. Στο πλαίσιο του Maven, οι εξαρτήσεις δηλώνονται στο αρχείο Project Object Model (POM) του έργου. Κάθε εξάρτηση καθορίζει ένα εξωτερικό πακέτο ή βιβλιοθήκη που απαιτείται για τη σωστή εκτέλεση του έργου. Η κάθε εξάρτηση περιλαμβάνει τρία βασικά στοιχεία: το groupId, το artifactId και την έκδοση [20]. Το groupId προσδιορίζει την ομάδα που δημιούργησε τη βιβλιοθήκη, ενώ το artifactId είναι το όνομα της βιβλιοθήκης. Η έκδοση υποδεικνύει ποιά έκδοση βιβλιοθήκης πρέπει να χρησιμοποιηθεί. Η σωστή διαχείριση των εξαρτήσεων εξασφαλίζει ότι το έργο θα μπορεί να εκτελείται χωρίς σφάλματα λόγω έλλειψης απαραίτητων βιβλιοθηκών [21].

Για παράδειγμα, ας υποθέσουμε ότι θέλουμε να χρησιμοποιήσουμε τη βιβλιοθήκη *spring-core* της ομάδας *org.springframework* στην έκδοση 5.3.8. Μπορούμε να συμπεριλάβουμε αυτήν την εξάρτηση μέσα στο αρχείο POM μας ως εξής:

```
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-core</artifactId>
  <version>5.3.8</version>
</dependency>
```

### Σχήμα 2.3.1 Org.springframework Version 5.3.8

Με αυτόν τον τρόπο, δηλώνουμε ότι το έργο μας εξαρτάται στη βιβλιοθήκη *spring-core* της ομάδας *org.springframework* στην έκδοση 5.3.8. Η Maven θα διασφαλίσει ότι αυτή η έκδοσή θα ληφθεί στον κατάλληλο κατάλογο *lib* ή *target* του έργου μας, ώστε να είναι διαθέσιμη κατά τη διάρκεια της μεταγλώττισης. Έτσι όταν θέλουμε να την χρησιμοποιήσουμε στον κώδικα μας θα μπορούμε απλά να την καλέσουμε.

Στο πλαίσιο της Maven, η κατηγορία των βιβλιοθηκών αναφέρεται σε συλλογές κώδικα που παρέχουν συγκεκριμένες λειτουργίες ή δυνατότητες. Κάθε εξάρτηση που καθορίζεται στο αρχείο POM του Maven οδηγεί στη λήψη και ενσωμάτωση της αντίστοιχης βιβλιοθήκης στο έργο [22]. Οι βιβλιοθήκες αποτελούν ουσιαστικά σύνολα προγραμματιστικών εργαλείων που επιτρέπουν την ανάπτυξη λογισμικού με περισσότερη αποτελεσματικότητα και ταχύτητα. Παρά να δημιουργούνται νέες λειτουργίες από την αρχή κάθε φορά που απαιτείται κάτι συγκεκριμένο, οι προγραμματιστές μπορούν να χρησιμοποιήσουν προϋπάρχουσες βιβλιοθήκες που παρέχουν έτοιμες λύσεις για συγκεκριμένα προβλήματα. Μπορεί να πρόκειται για βιβλιοθήκες γραφικών χρήστη, βιβλιοθήκες διαχείρισης βάσεων δεδομένων, εργαλεία ασφάλειας, ή οποιοδήποτε άλλο είδος βιβλιοθήκης που καλύπτει τις απαιτήσεις του προγράμματος [23]. Ένα σημαντικό πλεονέκτημα των βιβλιοθηκών είναι ότι συνήθως περιλαμβάνουν τεκμηρίωση και παραδείγματα χρήσης που βοηθούν τους προγραμματιστές να τις αξιοποιούν σωστά. Αυτό επιτρέπει στους προγραμματιστές να

εξοικειωθούν με τα χαρακτηριστικά και τις δυνατότητες της βιβλιοθήκης καθώς παράλληλα να ενσωματώσουν με επιτυχία την βιβλιοθήκη στο πρόγραμμά τους [24].

## **2.4 Γιατί κάποιος χρήστης να χρησιμοποιήσει την Maven**

Αν κάποιος χρήστης αποφασίσει να χρησιμοποιήσει την Maven, θα επωφεληθεί από εργαλεία και λειτουργίες που αυτοματοποιούν τις διαδικασίες ανάπτυξης λογισμικού, εξοικονομώντας χρόνο και μειώνοντας τον κίνδυνο λαθών [25]. Μερικά από τα κύρια λάθη που βελτιώνει η Maven είναι η αναμιξογραφία των εκδόσεων των βιβλιοθηκών. Η Maven διαχειρίζεται τις εξαρτήσεις και επιτρέπει την εύκολη πρόσβαση σε συγκεκριμένες εκδόσεις βιβλιοθηκών, αποφεύγοντας έτσι τη σύγκρουση εκδόσεων. Επίσης, η κακή διαχείριση των εξαρτήσεων είναι ακόμη ένα λάθος που η Maven λύνει παρέχοντας ένα σαφές και οργανωμένο σύστημα διαχείρισης των εξαρτήσεων, βοηθώντας στην εγκατάσταση και ενημέρωσή τους [26]. Επιπλέον, η Maven διασφαλίζει τη συμβατότητα μεταξύ των εξαρτήσεων ενός έργου, βοηθώντας έτσι στην αποφυγή σφαλμάτων που μπορεί να προκύψουν λόγω μη συμβατών εκδόσεων. Ακόμη, παρέχει μια οργανωμένη δομή για τα έργα λογισμικού μέσω των αρχείων POM, κάνοντας τη διαχείριση έκδοσης και εξάρτησης απλή [27]. Τελευταίο αλλά εξίσου σημαντικό, η εκτεταμένη κοινότητα της Maven παρέχει υποστήριξη, τεκμηρίωση και πόρους για τους χρήστες, κάνοντας την επιλογή της Maven ακόμα πιο ελκυστική [28].

# Κεφάλαιο 3

## Απαιτήσεις

---

|                                |    |
|--------------------------------|----|
| 3.1 Απαιτήσεις συστήματος      | 9  |
| 3.2 Λειτουργικές απαιτήσεις    | 9  |
| 3.3 Μη λειτουργικές απαιτήσεις | 10 |

---

### 3.1 Απαιτήσεις συστήματος

Το σύστημα που παρουσιάζεται σε αυτό το έγγραφο αποτελεί ένα πλαίσιο για τη διαχείριση και την ανάκτηση πληροφοριών σχετικά με τα πακέτα λογισμικού που διατίθενται μέσω του Maven Repository. Η κύρια λειτουργία του συστήματος είναι να παρέχει στους χρήστες ένα εύχρηστο περιβάλλον για την εξερεύνηση, την αναζήτηση και την ανάκτηση πληροφοριών σχετικά με τα πακέτα της Maven. Οι λειτουργικές και μη λειτουργικές απαιτήσεις που απαιτούνται για την υλοποίηση αυτού του σκοπού περιγράφονται σε βάθος στις απαιτήσεις που ακολουθούν.

### 3.2 Λειτουργικές απαιτήσεις

Οι λειτουργικές απαιτήσεις, εστιάζουν στην περιγραφή των βασικών λειτουργιών του συστήματος [29]. Θα αναλυθούν οι δυνατότητες που προσφέρει το σύστημα για την αποθήκευση δεδομένων σε διάφορα format, καθώς και η πρόσβαση σε πληροφορίες σχετικά με έργα και πακέτα Maven, αναδεικνύοντας τον ρόλο τους στην βελτίωση της αποδοτικότητας και της αλληλεπίδρασης χρηστών με το σύστημα.



- Αρχικά, το σύστημα ενώνει την γλώσσα προγραμματισμού Java με την βάση δεδομένων για να μπορούμε να μεταφέρουμε τα δεδομένα από τα πακέτα στην βάση μας [30].
- Έπειτα, η βάση ενώνεται με την σελίδα που κατασκευάσαμε για να μεταφέρει τα δεδομένα στον χρήστη όταν της ζητηθεί.
- Επίσης, το σύστημα παρέχει στον χρήστη τη δυνατότητα αναζήτησης πακέτων Maven βάσει διαφόρων κριτηρίων που αυτός θα επιλέξει [31]. Τα αποτελέσματα της αναζήτησης παρουσιάζονται στην επόμενη σελίδα με λεπτομερή πληροφορία για κάθε πακέτο, επιτρέποντας στον χρήστη να μελετήσει εύκολα και γρήγορα τα επιθυμητά πακέτα.
- Επιπλέον, ο χρήστης μπορεί να διαχειριστεί τις εξαρτήσεις ενός πακέτου, εντοπίζοντας τα πακέτα που εξαρτώνται από αυτό και το αντίστροφο.
- Επιπρόσθετα, τα αποτελέσματα αναζήτησης μπορούν να αποθηκευτούν σε διάφορες μορφές αποθήκευσης, για μελλοντική ανάκτηση και χρήση από τον χρήστη [32].
- Τέλος, παρέχεται η δυνατότητα πλοήγησης στα αποτελέσματα αναζήτησης μέσω κουμπιών προβολής προηγούμενης και επόμενης σελίδας.

### 3.3 Μη Λειτουργικές απαιτήσεις

Σημαντικοί παράγοντες περιλαμβάνονται και στις μη λειτουργικές απαιτήσεις του συστήματος όπως η απόδοση, η ευκολία συντήρησης, η χρηστικότητα και η διαθεσιμότητα, που εξασφαλίζουν την απρόσκοπτη λειτουργία του συστήματος κατά τη διάρκεια της χρήσης του από τους χρήστες [33].

- Το σύστημα είναι αποδοτικό διότι ανταποκρίνεται γρήγορα στον χρήστη, περίπου στα τρία δευτερόλεπτα για κάθε αναζήτηση. Επίσης παίρνει τα δεδομένα των πακέτων από την Maven Central Repository περίπου στα 5 δευτερόλεπτα το πιο μεγάλο πακέτο.
- Είναι προσβάσιμο συνεχώς στις πληροφορίες των πακέτων Maven προτρέποντας τον χρήστη να μελετά συνέχεια τα δεδομένα που χρειάζεται χωρίς περιορισμούς.
- Το σύστημά παρέχει επίσης ασφάλεια, καθώς κάθε φορά που ένας χρήστης έχει πρόσβαση στη σελίδα και αναζητά δεδομένα, τα δεδομένα αυτά είναι ορατά και

προσβάσιμα μόνο από αυτόν τον συγκεκριμένο χρήστη [34]. Υπάρχει εγγύηση ότι δεν υπάρχει κίνδυνος παρέμβασης ή τροποποίησης των δεδομένων από άλλους χρήστες.

- Επιπλέον, είναι χρηστικό αφού παρέχει στον χρήστη πολύ λίγα κουμπιά και πολλά πεδία αναζήτησης. Με αυτό τον τρόπο επιτρέπει στον χρήστη να εντοπίσει συγκεκριμένα τα πακέτα που χρειάζεται, χωρίς να του εμφανίζονται αμέτρητα πακέτα και να ψάχνει ένα-ένα για να βρει το δικό του. Αυτό βοηθά τον χρήστη να κατανοήσει την λειτουργία του συστήματος και πιο γρήγορα.
- Επιπρόσθετα, παρέχεται διαθεσιμότητα στο σύστημα [35] αφού οι χρήστες μπορούν να αναζητήσουν πακέτα ανά ώρα, χωρίς να αντιμετωπίσουν διακοπές στην υπηρεσία οποιαδήποτε στιγμή της ημέρας.
- Έπειτα, υπάρχει ευελιξία στο σύστημα. Η δυνατότητα ενημέρωσης των δεδομένων όταν αλλάζουν τα rom xml αρχεία επιτρέπει στο σύστημα να προσαρμόζεται αυτόματα σε νέες πληροφορίες χωρίς παρέμβαση του χρήστη [36]. (Πρέπει να τρέξει η Java για να ενημερωθούν τα δεδομένα). Αυτό επιτρέπει τη διατήρηση της λειτουργικότητας του συστήματος σε μεταβαλλόμενα περιβάλλοντα. Ακόμη, η ευελιξία είναι σημαντική στη δυνατότητα επιλογής του τρόπου αποθήκευσης των δεδομένων από τους χρήστες (CSV, XML ή JSON), καθώς και τη δυνατότητα παροχής διαφορετικών τρόπων αναζήτησης και προβολής των πακέτων.
- Τέλος, μια ακόμη μη λειτουργική απαίτηση που υπάρχει είναι η αξιοπιστία, αφού υπάρχει σωστή αποθήκευση δεδομένων στην βάση δεδομένων και διατηρεί τα δεδομένα του συστήματος με ακεραιότητα [37].

# Κεφάλαιο 4

## Σχεδίαση συστήματος

---

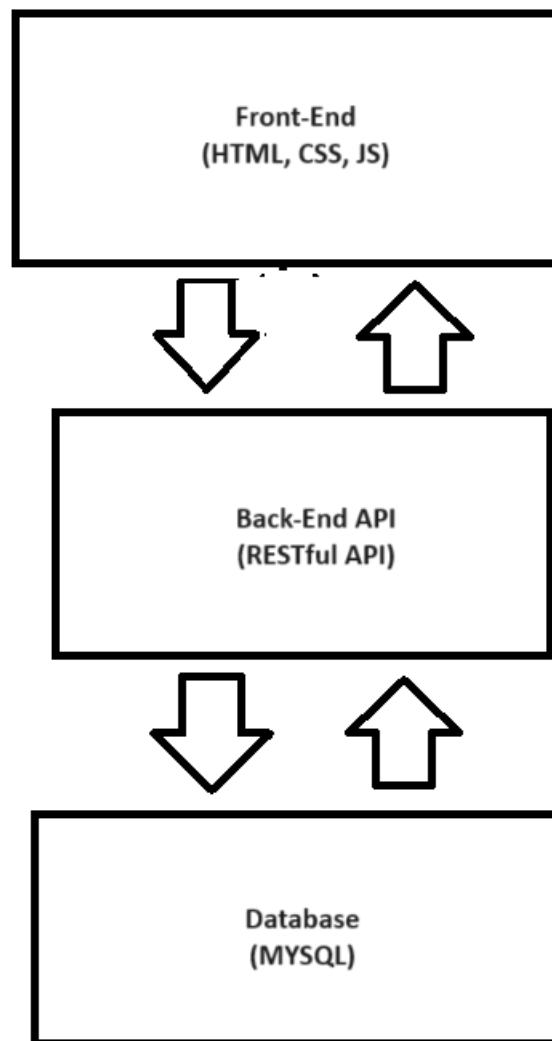
|         |                               |    |
|---------|-------------------------------|----|
| 4.1     | Αρχιτεκτονική συστήματος      | 13 |
| 4.1.1   | Υψηλού Επιπέδου Αρχιτεκτονική | 13 |
| 4.1.2   | Σχεδίαση Βάσης Δεδομένων      | 15 |
| 4.2     | Σχεδίαση συστήματος           | 16 |
| 4.3     | Front-end Development         | 17 |
| 4.3.1   | HTML5                         | 18 |
| 4.3.2   | CSS                           | 19 |
| 4.3.3   | JavaScript                    | 19 |
| 4.3.4   | BOOTSTRAP                     | 20 |
| 4.4     | Back-end Development          | 21 |
| 4.4.1   | MYSQL                         | 21 |
| 4.4.2   | PHP                           | 22 |
| 4.4.3   | JSON                          | 24 |
| 4.4.4   | JDBC                          | 24 |
| 4.4.5   | JAVA                          | 25 |
| 4.4.5.1 | Jsoup                         | 25 |
| 4.4.5.2 | Javax.xml                     | 26 |
| 4.4.5.3 | org.w3c.dom                   | 27 |
| 4.4.5.4 | org.xml.sax                   | 27 |

---

## 4.1 Αρχιτεκτονική του Συστήματος

### 4.1.1 Υψηλού Επιπέδου Αρχιτεκτονική

Παρακάτω παρουσιάζεται το διάγραμμα αρχιτεκτονικής του συστήματός μας, το οποίο απεικονίζει τα κύρια στοιχεία και τις αλληλεπιδράσεις τους.



Σχήμα 4.1.1.1 Διάγραμμα Αρχιτεκτονικής

## Περιγραφή των Κύριων Στοιχείων:

**Front-End:** Το front-end αποτελείται από HTML, CSS και JavaScript [38] και είναι υπεύθυνο για την εμφάνιση και την αλληλεπίδραση με τον χρήστη. Στέλνει αιτήματα στο Back-End API και εμφανίζει τα δεδομένα που λαμβάνει.

**Back-End API:** Το back-end παρέχει RESTful APIs [39] που εξυπηρετούν αιτήματα από το front-end, εκτελεί αιτήματα στη βάση δεδομένων και επιστρέφει τα δεδομένα στο front-end.

**Database:** Η βάση δεδομένων MySQL αποθηκεύει τα δεδομένα των πακέτων, τις εκδόσεις τους και τις εξαρτήσεις. Επεξεργάζεται αιτήματα CRUD [40] από το back-end.

Τα αιτήματα CRUD είναι μια συντομογραφία που αναφέρεται σε τέσσερις βασικές λειτουργίες που μπορούν να εκτελεστούν σε μια βάση δεδομένων ή σε οποιαδήποτε άλλη μορφή διαχείρισης δεδομένων [41]. Αυτές οι λειτουργίες είναι οι εξής:

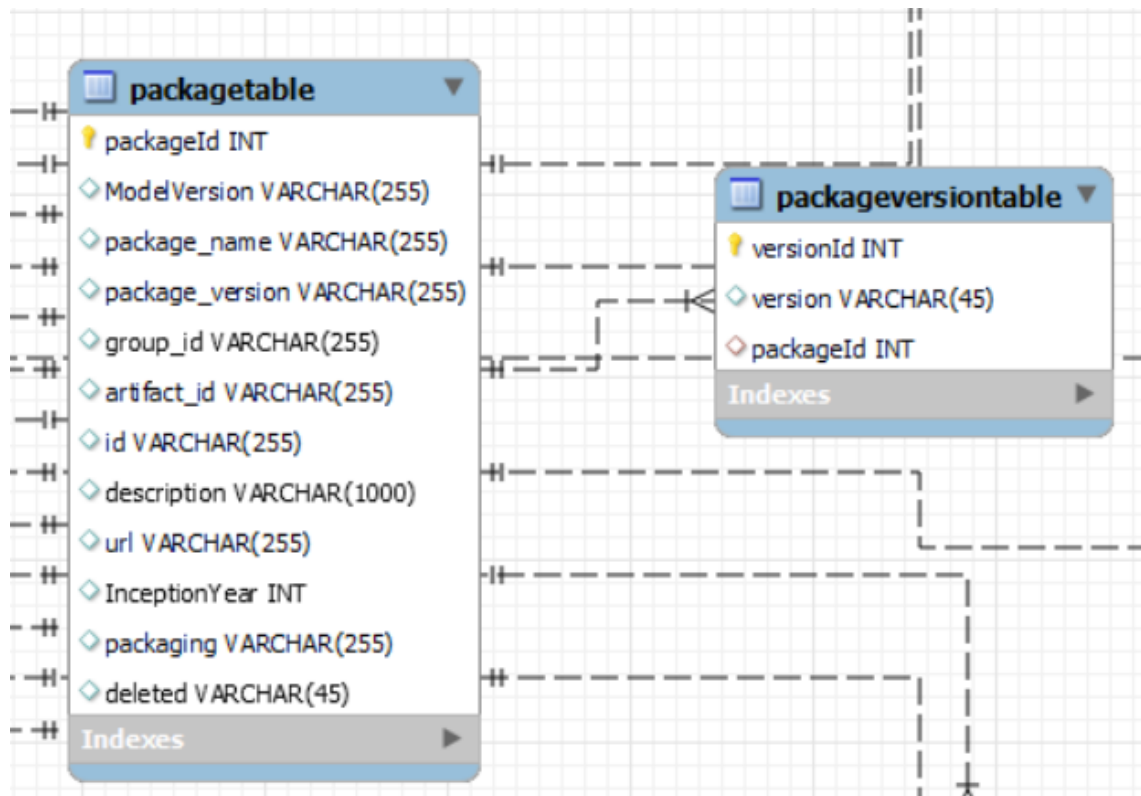
**Create** (Δημιουργία): Αφορά τη δημιουργία νέων εγγραφών στη βάση δεδομένων. Κατά το αίτημα δημιουργίας, νέα δεδομένα εισάγονται στη βάση δεδομένων.

**Read** (Ανάγνωση): Αναφέρεται στην ανάκτηση δεδομένων από τη βάση δεδομένων. Τα αιτήματα ανάγνωσης επιστρέφουν υπάρχοντα δεδομένα από τη βάση δεδομένων.

**Update** (Ενημέρωση): Χρησιμοποιείται για την τροποποίηση υπάρχοντων εγγραφών στη βάση δεδομένων. Κατά το αίτημα ενημέρωσης, τα υπάρχοντα δεδομένα τροποποιούνται με νέα δεδομένα.

**Delete** (Διαγραφή): Αφορά τη διαγραφή υπάρχοντων εγγραφών από τη βάση δεδομένων. Κατά το αίτημα διαγραφής, εγγραφές αφαιρούνται από τη βάση δεδομένων.

#### 4.1.2 Σχεδίαση Βάσης Δεδομένων



Σχήμα 4.1.2.1 ER Diagram

Παρουσιάζεται το ER διάγραμμα στο Σχήμα 4.1.2.1, όπου απεικονίζει τις οντότητες της βάσης δεδομένων και τις σχέσεις τους όπως η οντότητα “packagetable” και η οντότητα “packageVersionTable”. Σε αυτές τις οντότητες αποθηκεύονται οι τελευταίες εκδόσεις των πακέτων ώστε οι χρήστες να μπορούν να δουν την πιο πρόσφατη έκδοση.

#### Περιγραφή Οντοτήτων και Σχέσεων:

Στο ER Diagram, όλες οι οντότητες συνδέονται μεταξύ τους μέσω κάποιου Foreign Key [42]. Το Κύριο κλειδί που όλες οι οντότητες χρησιμοποιούν είναι το PackageId το οποίο είναι το κλειδί της πιο βασικής μας οντότητας PackageTable.

Δύο από τις πιο βασικές οντότητες του διαγράμματος είναι [43] :

**Packagetable:** Αυτή η οντότητα αντιπροσωπεύει τα πακέτα Maven και περιέχει πληροφορίες όπως το όνομα, την περιγραφή και το μοναδικό ID του πακέτου. Έχει σαν primary Key το packageId στο οποίο όλοι οι πίνακες συνδέονται με αυτό ώστε να γνωρίζουμε πιο ακριβώς πακέτο αντιστοιχεί με τα στοιχεία όλων των πινάκων.

**PackageVersionTable:** Αυτή η οντότητα περιέχει τις διάφορες εκδόσεις ενός πακέτου, συνδέεται με την οντότητα 'Package' μέσω ενός ξένου κλειδιού. Με αυτό τον τρόπο αν το packageId για παράδειγμα είναι 1 τότε στον packageVersionTable όλες οι εκδόσεις με packageId=1 αντιστοιχούν σε αυτό το πακέτο.

## 4.2 Σχεδίαση συστήματος

Αρχικά, το σύστημα μπορεί να πάρει τα δεδομένα από τα pom xml αρχεία με χρήση του Jsoup και να τα αποθηκεύσει στην βάση δεδομένων με χρήση του JDBC σε διάφορους πίνακες. Το JDBC παρέχει σταθερότητα και συμβατότητα με τις περισσότερες βάσεις δεδομένων [44]. Υπάρχουν πίνακες όπως ο packagetable στον οποίο μπαίνουν τα δεδομένα του πακέτου όπως η περιγραφή, το όνομα και το id καθώς και πίνακες όπως τον packageVersionTable όπου εδώ αποθηκεύονται οι πιο πρόσφατες εκδόσεις των πακέτων ώστε ο χρήστης να βλέπει την πιο πρόσφατη έκδοση. Επίσης, το σύστημα μπορεί να κάνει update τα δεδομένα όταν αυτά αλλάξουν στα pom xml αρχεία, έτσι ώστε να έχουμε πάντα τα νεότερα δεδομένα για τον χρήστη. Επιπρόσθετα, τα δεδομένα της βάσης δεδομένων όπως το username, url και password [45] αποθηκεύονται σε ένα ξεχωριστό αρχείο έτσι ώστε να μην μπορεί κάποιος άγνωστος να κάνει inspect τον κώδικα και να έχει πρόσβαση στην βάση δεδομένων.

Επιπλέον, το σύστημα παρέχει στον χρήστη τη δυνατότητα να αναζητήσει πακέτα Maven με βάση οποιαδήποτε κριτήρια τον ενδιαφέρουν, όπως το όνομα του συγγραφέα, την έκδοση, ή το groupId. Τα αποτελέσματα της αναζήτησης εμφανίζονται στο πάνω μέρος της σελίδας με τον αριθμό πακέτων που βρέθηκαν για το συγκεκριμένο πεδίο που

εισήγαγε ο χρήστης παρουσιάζοντας τα πακέτα με πάρα πολλά στοιχεία όπως το όνομα του πακέτου, η έκδοση του, το group ID του κ.λπ. Αυτό επιτρέπει στον χρήστη, να μελετήσει τα πακέτα που τον ενδιαφέρουν πιο εύκολα αφού θα έχει μια πλήρη γκάμα πληροφοριών για το κάθε πακέτο.

Ακόμη, ο χρήστης έχει την δυνατότητα να μελετήσει για τις εξαρτήσεις κάποιου πακέτου με το “dependsOn” και το “dependedOn” βρίσκοντας έτσι από ποια άλλα πακέτα εξαρτάται το πακέτο του και το αντίθετο. Τα δεδομένα μπορούν να αποθηκευτούν σε μορφή CSV, XML ή JSON, παρέχοντας έτσι ευελιξία στον τρόπο αποθήκευσης και χρήσης τους από τον χρήστη. Παράλληλα, ο χρήστης έχει τη δυνατότητα να περιηγηθεί στα αποτελέσματα αναζήτησης χρησιμοποιώντας τα κουμπιά προβολής προηγούμενης “Prev” και επόμενης σελίδας “Next”, καθώς και να μεταβεί σε συγκεκριμένη σελίδα πατώντας το κουμπί ‘Go’, επιτρέποντάς του έτσι να εξερευνήσει τα δεδομένα με άνεση και αποτελεσματικότητα.

### **4.3 Front-end Development**

Το front-end development αφορά τη δημιουργία του χρηστικού περιβάλλοντος μιας ιστοσελίδας ή εφαρμογής, εστιάζοντας στην εξωτερική εμφάνιση και λειτουργία του συστήματος, καθώς και στον τρόπο αλληλεπίδρασης με τον χρήστη. Ποιο κάτω θα αναφερθούν λεπτομερώς οι τεχνολογίες που χρησιμοποιήσαμε για το σύστημα.



### 4.3.1 HTML5

Το HTML5 είναι η πέμπτη έκδοση της γλώσσας σήμανσης HTML και παρέχει αρκετές πρόσθετες δυνατότητες σε σχέση με τις προηγούμενες εκδόσεις. Νέες σημασίες και στοιχεία <header>, <footer>, <nav>, <article>, <section> [46] που καθιστούν τη δομή του κώδικα HTML πιο κατανοητή, περιλαμβάνονται στα κύρια χαρακτηριστικά της HTML5. Επιπλέον, η HTML5 επιτρέπει την άμεση ενσωμάτωση εικόνων, βίντεο και μουσικής χωρίς την απαίτηση για πρόσθετα προγράμματα περιήγησης [47]. Η HTML5 ήταν χρήσιμη για την δημιουργία της φόρμας καθώς και των πεδίων όπου μπορεί ο χρήστης να τοποθετήσει εκεί αυτά που επιθυμεί. Επίσης ήταν σημαντική για τα κουμπιά της σελίδας τα οποία διαχειρίζεται ο χρήστης καθώς και για την εικόνα[48]. Αντίθετα, η Haml είναι μια διαφορετική προσέγγιση στη σήμανση ιστοσελίδων, με λιγότερη χρήση χαρακτήρων και πιο συνοπτική σύνταξη. Παρόλο που η Haml μπορεί να προσφέρει κάποια πλεονεκτήματα σε ορισμένες περιπτώσεις, η HTML παραμένει η πιο διαδεδομένη και σταθερή επιλογή για τη δημιουργία ιστοσελίδων. Αυτός είναι και ο βασικότερος λόγος που χρησιμοποίησα HTML5 για τον ‘σκελετό’ της σελίδας μας.



Uploaded by [Solarize Pictures](#)

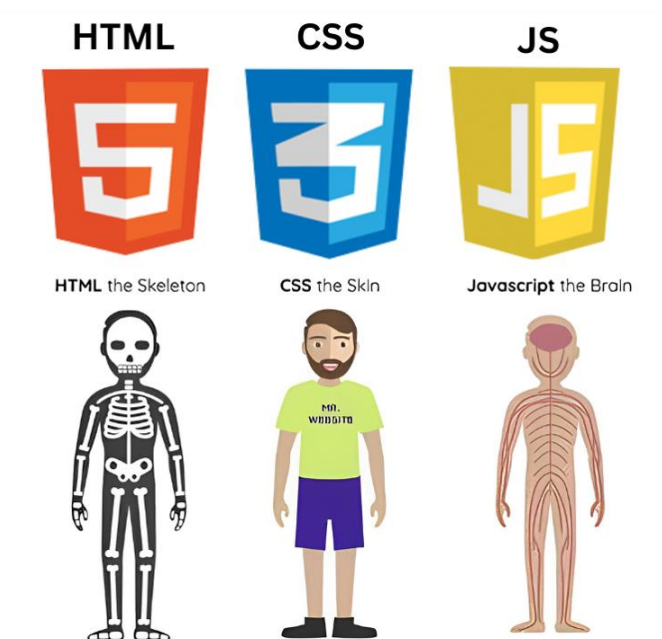
**Σχέδιο 4.1.1.1 HTML5**

### 4.3.2 CSS

Το CSS (Cascading Style Sheets) είναι μια γλώσσα που χρησιμοποιείται για τη μορφοποίηση του περιεχομένου ενός εγγράφου HTML ή XML [49]. Η κύρια λειτουργία του CSS είναι να δίνει στυλ και εμφάνιση στα στοιχεία της ιστοσελίδας - όπως κείμενο, φόντο, περιγραφικές γραμμές, και περιθώρια. Οι επιλογές στο CSS χρησιμοποιούνται για την επιλογή των στοιχείων HTML στα οποία θα εφαρμοστούν τα στυλ. Συγκεκριμένα στη σελίδα μας χρησιμοποιήσαμε αρκετή CSS αφού μας βοήθησε να στοιχίσουμε τα πεδία μας, να βάλουμε χρώμα στους τίτλους καθώς και να διαμορφώσουμε το πίσω μέρος της σελίδας μας. Επίσης, μας βοήθησε για να δώσουμε το κατάλληλο μέγεθος στην εικόνα μας στην πάνω αριστερά γωνία ώστε να διαμορφώνεται όμορφα στη σελίδα. Η χρήση της CSS είναι πολύ σημαντική για μια σελίδα γιατί την αναδιαμορφώνει και την κάνει πιο ελκυστική για τους επισκέπτες, προσελκύοντας έτσι περισσότερους χρήστες με ενδιαφέρον.

### 4.3.3 JavaScript

Η JavaScript είναι μια δυναμική γλώσσα προγραμματισμού που χρησιμοποιείται κυρίως για τη δημιουργία διαδραστικών ιστοσελίδων. Είναι ο εγκέφαλος της ιστοσελίδας, επιτρέποντάς την ενσωμάτωση δυναμικής λειτουργικότητας όπως η αλληλεπίδραση με τον χρήστη. Μπορεί να χρησιμοποιηθεί για τη δημιουργία ενός ευρέος φάσματος, από απλές διαδραστικές φόρμες έως πολύπλοκες εφαρμογές web [50]. Στη δική μας περίπτωση, η JavaScript ήταν πάρα πολύ σημαντική αφού μας βοήθησε με τα κουμπιά και την λειτουργία τους. Τα δύο βασικά κουμπιά της σελίδας μας είναι το 'search' το οποίο μόλις το πατήσει ο χρήστης αναλαμβάνει να πάει τον χρήστη στην επόμενη σελίδα καθώς και να εμφανίσει τα πακέτα τα οποία ο χρήστης ζήτησε. Το δεύτερο βασικό κουμπί είναι το 'back' το οποίο είναι εξίσου σημαντικό, αφού αν ο χρήστης ξεχάσει να βάλει κάποιο πεδίο στην φόρμα αναζήτησης ή πατήσει κατά λάθος το κουμπί 'search' απευθείας τον οδηγεί στην προηγούμενη σελίδα για να τοποθετήσει ξανά εκεί ότι επιθυμεί.



Σχήμα 4.3.3.1

[Anass Essadikine](#) Full Stack Developer | WordPress Developer | PowerBI Specialist  
Published Jan 17, 2024

#### 4.3.4 BOOTSTRAP

Το Bootstrap είναι ένα πλαίσιο εργασίας (framework) CSS που σχεδιάστηκε για να κάνει τη δημιουργία βελτιωμένων ιστότοπων και εφαρμογών απλούστερη και ταχύτερη. Το Bootstrap παρέχει ένα σύνολο προκαθορισμένων στυλ, συστατικών και διεπαφών χρήστη (UI components) [51] που μπορούν να χρησιμοποιηθούν για τη δημιουργία ενός ευέλικτου και εξειδικευμένου ιστότοπου. Η ταχύτητα ανάπτυξης, η ανταπόκριση, η στυλιστική συνοχή, η ευελιξία και η κοινότητα και η υποστήριξη [52] είναι μερικά από τα βασικά οφέλη του Bootstrap. Συγκεκριμένα, στο σύστημα μας το Bootstrap ήταν πολύ χρήσιμο αφού μας βοήθησε αρκετά με την στοίχιση των κουμπιών ώστε να φαίνονται όλα ομοιόμορφα καθώς και στο χρώμα των κουμπιών αφού έχουμε τοποθετήσει ένα πολύ συγκεκριμένο χρώμα στα κουμπιά μας ώστε να φαίνονται όμορφα στον χρήστη.

## 4.4 Back-end Development

Σε αντίθεση με το Front-end Development, το back-end development αναφέρεται στην ανάπτυξη του server-side τμήματος μιας ιστοσελίδας ή μιας εφαρμογής [53]. Ενώ το front-end εστιάζει στη δημιουργία του χρήσιμου περιβάλλοντος που βλέπει και αλληλεπιδρά με αυτό ο χρήστης, το back-end εστιάζει στη διαχείριση δεδομένων, την επικοινωνία με τη βάση δεδομένων, τον υπολογισμό και την επεξεργασία των αιτημάτων των χρηστών, και την παραγωγή του απαραίτητου περιεχομένου που στέλνεται στο front-end για προβολή.

### 4.4.1 MYSQL

Στον κόσμο του web και των εφαρμογών λογισμικού, η MySQL είναι μια δημοφιλής σχεσιακή βάση δεδομένων. Ανάμεσα στα βασικά στοιχεία της περιλαμβάνεται η δημιουργία και διαχείριση βάσεων δεδομένων οι οποίες παρέχουν δεδομένα σχετικά με τους ιστότοπους ή τις εφαρμογές. Η MySQL χρησιμοποιεί τη γλώσσα ερωτημάτων SQL (Structured Query Language) [54] που επιτρέπει τη δημιουργία, την ενημέρωση, τη διαγραφή και την ανάκτηση δεδομένων. Επίσης προσφέρει λειτουργίες ασφαλείας για την προστασία δεδομένων, ελέγχοντας τα δικαιώματα πρόσβασης και επιβάλλοντας όρια ασφαλείας. Τέλος, η MySQL διευκολύνει τις συναλλαγές, διασφαλίζοντας τη συνέπεια και την ακεραιότητα της βάσης δεδομένων από τα αιτήματα της php [55]. Η MYSQL είναι η βάση δεδομένων που χρησιμοποιήσα για την αποθήκευση δεδομένων από την Maven Central Repository. Συγκεκριμένα, η βάση δεδομένων μας αποτελείται από αρκετούς πίνακες όπου ο κάθε πίνακας έχει τα δικά του δεδομένα. Όλοι οι πίνακες συνδέονται μεταξύ τους μέσω ενός id, του packageId όπου δείχνει σε πιο πακέτο βρίσκεται το κάθε στοιχείο του πίνακα.

Για παράδειγμα:

| packageId | ModelVersion | package_name                | package_version | group_id   | artifact_id |
|-----------|--------------|-----------------------------|-----------------|------------|-------------|
| 1         | 4.0.0        | HTTPClient.HTTPClient       | 0.3-3           | HTTPClient | HTTPClient  |
| 2         | 4.0.0        | Abbot Java GUI Test Library | 1.4.0           | abbot      | abbot       |

**Σχήμα 4.2.1.1 Πίνακας PackageTable**

Εδώ παρουσιάζεται το packageId “No.2” όπου περιέχει το ModelVersion(4.0.0), το package\_name (Abbot Java GUI Test Library) και τα package version, group\_id, artifact\_id ανάλογα.

| licenseId | packageId | name | url                                                                                                 |
|-----------|-----------|------|-----------------------------------------------------------------------------------------------------|
| 228497    | 2         | EPL  | <a href="https://www.eclipse.org/legal/epl-v10.html">https://www.eclipse.org/legal/epl-v10.html</a> |

**Σχήμα 4.2.1.2 Πίνακας licenceTable**

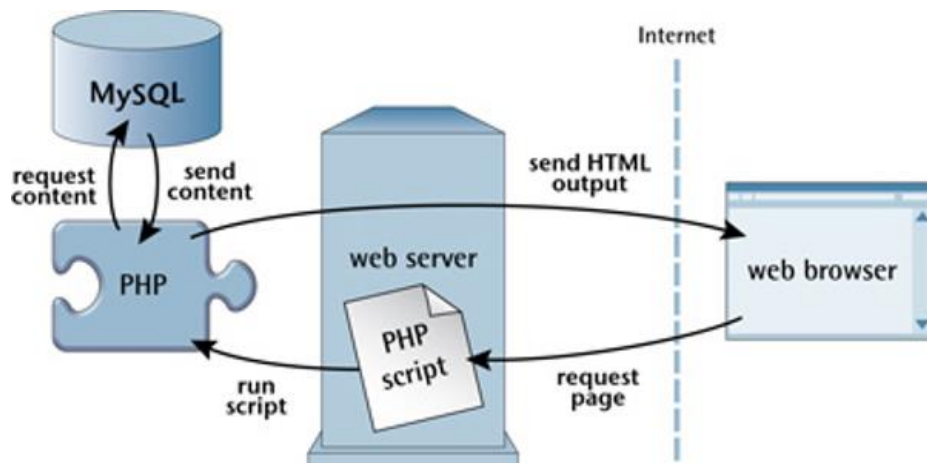
Ο πίνακας licenceTable παρέχει το ίδιο packageId με αυτό στο packageTable(Σχήμα 4.2.1.1). Αυτό σημαίνει ότι τα δεδομένα και στα δύο σχήματα περιέχονται μέσα σε ένα pom xml αρχείο στο Maven Central Repository.

## 4.4.2 PHP

Η PHP (Hypertext Preprocessor) είναι μια δημοφιλής γλώσσα προγραμματισμού σχεδιασμένη ειδικά για την ανάπτυξη δυναμικών ιστοσελίδων και εφαρμογών web. Η PHP είναι μια ευέλικτη και εύκολη στην εκμάθηση γλώσσα προγραμματισμού που μπορεί να χρησιμοποιηθεί για μια ευρεία γκάμα εφαρμογών, απο βασικές ιστοσελίδες έως περίπλοκες δυναμικές πλατφόρμες. Μπορεί να ενσωματωθεί εύκολα σε HTML, επιτρέποντας την ανάπτυξη δυναμικών ιστοσελίδων με ενεργά περιεχόμενα [56]. Η εκτέλεση κώδικα PHP από την πλευρά του διακομιστή (server-side), προσφέρει διαχείρισης δεδομένων και ασφάλειας. Η εύκολη χρήση της PHP διαθέτει πολλές

ενσωματωμένες λειτουργίες για τη διαχείριση αρχείων, την αλληλεπίδραση με βάσεις δεδομένων και τον χειρισμό δεδομένων. Η PHP είναι μια εύκολη στην εκμάθηση και ευέλικτη γλώσσα προγραμματισμού που μπορεί να χρησιμοποιηθεί για μια ευρεία γκάμα εφαρμογών, από απλές ιστοσελίδες μέχρι περίπλοκες δυναμικές πλατφόρμες [57]. Η χρήση του PHP ήταν αναγκαία για να μπορούμε να λαμβάνουμε δεδομένα από τη βάση δεδομένων και να τα εμφανίζουμε στην σελίδα μας για την προσωπική χρήση του χρήστη.

Πιο κάτω υπάρχει το σχήμα που δείχνει πως λειτουργεί ακριβώς το php.



**What is PHP? [https://selliliar.live/product\\_details/53307669.html](https://selliliar.live/product_details/53307669.html)**

#### **Σχήμα 4.4.2.1 Σχεδιάγραμμα λειτουργίας του PHP.**

*Στην πρώτη φάση ενώνουμε την σελίδα με την php, και την php με την βάση δεδομένων. Μετά τρέχουμε την php ώστε να πάρει όλα τα δεδομένα που χρειαζόμαστε από την βάση και να μας επιστρέψει όλα τα πακέτα που πρέπει ανάλογα. Στο τέλος, αφού επιστρέψουν τα πακέτα από την βάση δεδομένων αποθηκεύονται στην php και η php τα στέλνει στην σελίδα για display. Μετά από αυτή την διαδικασία ο χρήστης έχει την πρόσβαση να εντοπίσει τα πακέτα που αναζητεί.*

#### 4.4.3 JSON

Το JavaScript Object Notation είναι ένας απλός, ευανάγνωστο πρότυπο ανταλλαγής δεδομένων που χρησιμοποιείται στην ανάπτυξη λογισμικού [58]. Αν και αρχικά αναπτύχθηκε στο πλαίσιο της JavaScript, το JSON έχει γίνει ένας δημοφιλής μορφότυπος δεδομένων λόγω της απλότητας του και της προσαρμοστικότητας του. Οι πρωταρχικές του ιδιότητες συμπεριλαμβάνουν την απλότητα, την ανθρώπινη αναγνωσιμότητα, την ευελιξία, την αποτελεσματική μεταφορά δεδομένων [59] και την υποστήριξη από τη JavaScript. Λόγω της απλής σύνταξης και της ευκολίας κατανόησής του, το JSON είναι μια δημοφιλή μορφή για τη μεταφορά δεδομένων. Οι χρήστες μπορούν εύκολα να κατανοήσουν τα δεδομένα JSON χωρίς τη χρήση εξειδικευμένου εξοπλισμού.

Το JSON υποστηρίζει διάφορους τύπους δεδομένων, όπως αριθμούς, συμβολοσειρές, λογικές τιμές, πίνακες και αντικείμενα, κάνοντάς το κατάλληλο για μια ευρεία γκάμα εφαρμογών [60]. Επίσης είναι απλό στη συμπίεση, γεγονός που διευκολύνει τη γρήγορη μεταφορά δεδομένων σε διάφορα συστήματα. Τέλος, το JSON είναι προεγκατεστημένο με JavaScript και χρησιμεύει ως προεπιλεγμένη μορφή δεδομένων για τη μεταφορά δεδομένων μεταξύ PHP και JavaScript. Στο σύστημα μας, χρησιμοποιήσαμε το Json για να μετατρέψουμε τα δεδομένα που πήραμε από την βάση δεδομένων έτσι ώστε να είναι ευανάγνωστα για τον χρήστη όταν τα διαβάσει.

#### 4.4.4 JDBC

Το Java Database Connectivity (JDBC), είναι ένα πακέτο εργαλείων για τη σύνδεση βάσεων δεδομένων από προγράμματα Java. Χρησιμοποιώντας το JDBC, μπορούμε να εκτελέσουμε ερωτήματα σε διαφορετικά είδη βάσεων δεδομένων όπως η MySQL, η Oracle, ο SQL Server [61] και να μεταφέρουμε δεδομένα μεταξύ του προγράμματος Java και της βάσης δεδομένων. Η σύνδεση με βάση δεδομένων καθιστά δυνατή τη δημιουργία εφαρμογών Java που συνδέονται με διαφορετικές βάσεις δεδομένων. Επίσης, μπορούμε να εκτελέσουμε ερωτήματα SQL στις βάσεις δεδομένων και να λάβουμε απαντήσεις μέσω του JDBC. Παράλληλα, οι δυνατότητες διαχείρισης σύνδεσης βάσης δεδομένων που προσφέρει η JDBC περιλαμβάνουν μειωμένο, κλειστό και χειρισμό σφαλμάτων

[62]. Το JDBC, μας βοήθησε για να συνδέσουμε την βάση δεδομένων μας με την Java ώστε τα δεδομένα να μπορούν να μεταφερθούν και να τοποθετηθούν στην βάση δεδομένων μας απλά και αποτελεσματικά.

#### 4.4.5 Java

Η Java είναι μία γλώσσα προγραμματισμού που χρησιμοποιείται για την ανάπτυξη στιγμιαίων, έτοιμων διαδικασιών και εφαρμογών που θα λειτουργήσουν σε ένα διαδικτυακό περιβάλλον [63]. Με τη Java, μπορείς να αναπτύξεις προγράμματα που τρέχουν σε οποιονδήποτε υπολογιστή, ανεξάρτητα από το λειτουργικό σύστημα ή τη διαδικτυακή σελίδα. Μαζί με ένα τεράστιο οικοσύστημα βιβλιοθηκών και εργαλείων για την ανάπτυξη εφαρμογών, περιλαμβάνει επίσης χαρακτηριστικά ασφαλείας που προστατεύουν τις εφαρμογές από ιούς και μια εύκολη στην εκμάθηση σύνταξη [64]. Συγκεκριμένα η Java, ήταν η γλώσσα προγραμματισμού που μας επέτρεψε να εισάγουμε δεδομένα από την Maven Central Repository στη βάση δεδομένων μας χρησιμοποιώντας το JDBC. Αρχικά ελέγχουμε αν υπάρχει οποιαδήποτε συμφωνία μεταξύ των <name> tags στο responseBody. Αν βρει ένα τέτοιο μοτίβο, η μέθοδος find() του matcher επιστρέφει true, δηλώνοντας ότι βρέθηκε ένα τέτοιο στοιχείο. Αντίστοιχα, αν δε βρει κάποια αντιστοίχιση με το pattern, τότε η find() επιστρέφει false, δηλώνοντας ότι δεν υπάρχει τέτοιο στοιχείο στο responseBody. Σε αυτήν την περίπτωση, εμφανίζεται το μήνυμα "Package number not found". Εάν το find() επιστρέψει true, το πρόγραμμα συνεχίζει με την εκτέλεση της μεθόδου insert Package, περνώντας τα δεδομένα που έχουν εξαχθεί από το responseBody. Αν επιστρέψει false, τότε η μέθοδος insert Package δεν καλείται, αφού δεν βρέθηκε καμία αντιστοίχιση και συνεπώς, δεν υπάρχουν δεδομένα για να εισαχθούν.

(Περισσότερες πληροφορίες για την διαδικασία μπορείτε να βρείτε στα Παραρτήματα Α,Β και Γ αντίστοιχα).



#### 4.4.5.1 Jsoup

Το Jsoup είναι μια βιβλιοθήκη προγραμματισμού στη γλώσσα Java που κυρίως χρησιμοποιείται για την ανάλυση, τη διαχείριση και την εξαγωγή πληροφοριών από HTML και XML δομές [65]. Με την χρήση του Jsoup, μπορούμε να επιλέξουμε στοιχεία σε μια σελίδα HTML, να ανακτήσουμε το περιεχόμενό τους και να το επεξεργαστούμε με Java [66]. Λαμβάνοντας υπόψη όλα τα πράγματα, το Jsoup επιτρέπει στους προγραμματιστές Java να αλληλεπιδρούν με δεδομένα HTML και XML, αυξάνοντας τόσο την παραγωγικότητα όσο και την απόλαυση της ανάπτυξης του προγραμματισμού.

Συγκεκριμένα στο πρόγραμμά μας, το Jsoup χρησιμοποιείται ιδιαιτέρως για να συνδεθεί σε μια ιστοσελίδα, την Maven Central Repository σε αυτήν την περίπτωση, να πάρει τα στοιχεία των συνδέσμων και να ανιχνεύσει τα στοιχεία των pom αρχείων για να μας παρέχει όλα τα δεδομένα που χρειαζόμαστε.

#### 4.4.5.2 Javax.xml

Η javax.xml είναι μια βιβλιοθήκη της Java που παρέχει κοινές κλάσεις και διεπαφές για τη διαχείριση XML στην Java [67]. Αυτή η βιβλιοθήκη περιλαμβάνει μια σειρά από μεθόδους και λειτουργίες για τον χειρισμό XML δεδομένων, συμπεριλαμβανομένης της ανάλυσης (parsing), της δημιουργίας, της τροποποίησης και της επεξεργασίας XML εγγράφων. Ορισμένες κύριες κλάσεις και διεπαφές που περιέχονται στη βιβλιοθήκη javax.xml περιλαμβάνουν το `DocumentBuilder` όπου χρησιμοποιείται για την ανάλυση υπαρχόντων εγγράφων XML ή για τη δημιουργία νέων από XML Document. Για την χρησιμοποίηση νέων αντικειμένων `DocumentBuilder` η χρήση του `DocumentBuilderFactory` είναι αναγκαία. Στο πλαίσιο της επεξεργασίας εγγράφων XML, το `XPath` χρησιμοποιεί εκφράσεις για να διατυπώσει και να εντοπίσει πληροφορίες σε ένα XML [68]. Το javax.xml είναι ένα βοηθητικό εργαλείο για τη διαχείριση XML σε προγράμματα Java και παρέχει πρακτικές μεθόδους για την ανάλυση, τη δημιουργία και την τροποποίηση XML εγγράφων. Συγκεκριμένα χρησιμοποιήσαμε το `XPath` της βιβλιοθήκης javax.xml για να λάβουμε συγκεκριμένα δεδομένα από τα Pom xml αρχεία

μας όπως το `group_id`, `artifact_id`, και το `version`. Αυτά τα συγκεκριμένα δεδομένα υπάρχουν επίσης και στον `parentTable` και υπήρξε πρόβλημα για τον λόγο ότι σε μερικά αρχεία τα `group_id`, `artifact_id`, και `version`, βρίσκονταν πριν τα `packageTable` δεδομένα και έβαζε τα δεδομένα στην βάση δεδομένων ανάποδα στους δύο πίνακες. Δηλαδή τα δεδομένα που έπρεπε να μουν στο `packageTable` τα έβαζε στο `parentTable` και τα δεδομένα που έπρεπε να τα βάλει στον `parentTable` τα έβαζε στον `packageTable`. Αυτός ήταν ο βασικός λόγος που χρησιμοποιήσαμε αυτήν την βιβλιοθήκη.

#### **4.4.5.3 org.w3c.dom**

Η βιβλιοθήκη Java `org.w3c.dom` προσφέρει τις διεπαφές (interfaces) και τις κλάσεις που ορίζονται από το World Wide Web Consortium (W3C) για τη διαχείριση XML εγγράφων στην πλατφόρμα Java [69]. Ανάμεσα στα κύρια στοιχεία και χαρακτηριστικά της `org.w3c.dom`, το κύριο στοιχείο είναι το `Document` το οποίο παρέχει πρόσβαση σε όλα τα στοιχεία και τους κόμβους κειμένου αντιπροσωπεύοντας ένα πλήρες XML. Η `org.w3c.dom` είναι σημαντική για το χειρισμό XML εγγράφων στη Java καθώς προσφέρει ένα σύνολο προτύπων και διεπαφών για την αποτύπωση της δομής ενός XML εγγράφου. Χρησιμοποιώντας τα στοιχεία που έχουν διαβαστεί, χρησιμοποιήσαμε το DOM για να δημιουργήσουμε ολόκληρο το δέντρο DOM στον κώδικα. Αυτό μας πρόσφερε τη δυνατότητα να ανακτήσουμε και να επιλέξουμε περισσότερα στοιχεία XML άμεσα με την χρήση του XPath [70] μετά τη διαδικασία ανάλυσης SAX (parsing) την οποία θα εξετάσουμε με περισσότερες λεπτομέρειες στη επόμενη ενότητα.

#### **4.4.5.4 org.xml.sax**

Ακόμη μια βιβλιοθήκη με την ονομασία `org.xml.sax`, χρησιμοποιεί το Simple API for XML (SAX) για τη διαχείριση και την επεξεργασία δεδομένων XML [71]. Το SAX και το Document Object Model (DOM) είναι τα δύο API στη γλώσσα Java, για την ανάπτυξη XML. Συγκεκριμένα, η βιβλιοθήκη `org.xml.sax` παρέχει κλάσεις και διεπαφές που επιτρέπουν την ανάλυση (parsing) των XML δεδομένων που βασίζεται σε SAX. Τα XML δεδομένα αναλύονται σε συμβάντα, όπως το άνοιγμα ενός στοιχείου, το κλείσιμο

ενός στοιχείου, και η εύρεση κειμένου χρησιμοποιώντας SAX [72]. Αυτή η προσέγγιση ανάλυσης είναι διαφορετική από το DOM, όπου το XML διαχειρίζεται ως δέντρο αντικειμένων.

Αν και το SAX είναι αποδοτικό στο να διαβάζει μεγάλα XML αρχεία σειριακά, χειρίζεται τα δεδομένα XML ως stream αντί να τα αποθηκεύει εξ ολοκλήρου στη μνήμη. Αυτό προσδιορίζει μετά την ανάγνωση του XML [73], πως θα υπάρξει δυσκολία στην πρόσβαση στοιχείων στο δέντρο DOM μέσω του κώδικα μου. Μετά την μελέτη του, μελετήσαμε το XML με το SAX και αναλύσαμε τα δεδομένα που μας ενδιαφέρουν. Χρησιμοποιήσαμε το DOM για να δημιουργήσουμε ένα πλήρες δέντρο DOM με βάση τα στοιχεία που έχουν ήδη διαβαστεί. Αυτό μας επέτρεψε να χρησιμοποιήσουμε το XPath για να επιλέξουμε, να επεξεργαστούμε, και να ανακτήσουμε εύκολα επιπλέον στοιχεία από το XML μετά τη διαδικασία του SAX parsing. Αυτός ο συνδυασμός είναι πολύ αποτελεσματικός και ευέλικτος για την επεξεργασία XML δεδομένων. Είναι εξίσου σημαντικό και στον κώδικα μας, καθώς το χρησιμοποιήσαμε γιατί με το SAX έχουμε λιγότερη κατανάλωση μνήμης. Συνολικά, αυτή η διαδικασία είναι αποδοτική και σωστή, επιτρέποντάς μας να εκμεταλλευτούμε τα πλεονεκτήματα τόσο του SAX (για την αποδοτική ανάγνωση του XML) όσο και του DOM (για την εύκολη επεξεργασία μετά τη διαβάθμιση).

# Κεφάλαιο 5

## Λειτουργίες Συστήματος

---

### 5.1 Περιγραφή Λειτουργιών του συστήματος

---

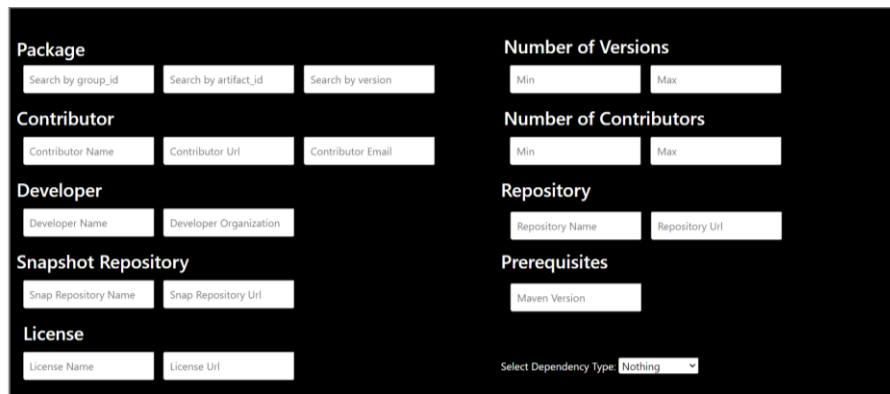
29

### 5.1 Περιγραφή Λειτουργιών του συστήματος

Το σύστημα υλοποιήθηκε στη γλώσσα προγραμματισμού Java και αποτελεί ένα εργαλείο αναζήτησης πακέτων και πληροφοριών πακέτων με την χρήση του Maven Central Repository. Επικεντρώνεται στην παροχή εύκολης και γρήγορης πρόσβασης σε πληροφορίες που αφορούν τα πακέτα και τις εξαρτήσεις τους, προσφέροντας ταυτόχρονα εργαλεία που διευκολύνουν τη διαχείρισή τους. Συγκεκριμένα χρησιμοποιήθηκαν εξωτερικές βιβλιοθήκες και εργαλεία όπως το Jsoup για την επεξεργασία των αρχείων pom.xml και το JDBC για την αποθήκευση των δεδομένων σε μια βάση δεδομένων. Μέσω μιας ολοκληρωμένης διεπαφής χρήστη, οι χρήστες μπορούν να πραγματοποιήσουν αναζητήσεις για πακέτα βασιζόμενοι σε διάφορα κριτήρια, να προβάλλουν λεπτομερείς πληροφορίες για τα πακέτα και τις εξαρτήσεις τους, και να διαχειρίζονται τη λίστα των επιλεγμένων πακέτων τους. Επιπλέον, παρέχεται η δυνατότητα εξατομικευμένων φίλτρων και επιλογών, συμπεριλαμβανομένων drop-down μενού για γρήγορη πρόσβαση σε κοινά χρησιμοποιούμενες λειτουργίες.

## Αναζήτηση Πακέτων:

Χρησιμοποιώντας τις διάφορες επιλογές που παρέχονται στην σελίδα, ο χρήστης μπορεί να αναζητήσει πακέτα με βάση διαφόρων κριτηρίων, όπως έκδοση (version), όνομα προγραμματιστή (developer name), όνομα συνεργάτη (contributor name) , και άδεια χρήσης (license).



The image shows a search interface with the following sections:

- Package**: Three input fields labeled "Search by group\_id", "Search by artifact\_id", and "Search by version".
- Contributor**: Three input fields labeled "Contributor Name", "Contributor Url", and "Contributor Email".
- Developer**: Two input fields labeled "Developer Name" and "Developer Organization".
- Snapshot Repository**: Two input fields labeled "Snap Repository Name" and "Snap Repository Url".
- License**: Two input fields labeled "License Name" and "License Url".
- Number of Versions**: Two input fields labeled "Min" and "Max".
- Number of Contributors**: Two input fields labeled "Min" and "Max".
- Repository**: Two input fields labeled "Repository Name" and "Repository Url".
- Prerequisites**: One input field labeled "Maven Version".
- Select Dependency Type**: A dropdown menu with "Nothing" selected.

Σχήμα 5.1.1

*Υπάρχουν διάφορα πεδία έτσι ώστε ο χρήστης να έχει την επιλογή να αναζητήσει ότι χρειάζεται.*

Ένα παράδειγμα αναζήτησης πακέτων παρουσιάζεται πιο κάτω:

**Package**  
abbot Search by artifact\_id Search by version

**Contributor**  
Contributor Name Contributor Url Contributor Email

**Developer**  
Gerard Davisonr Developer Organization

**Snapshot Repository**  
Snap Repository Name Snap Repository Url

**License**  
License Name License Url

**Number of Versions**  
Min Max

**Number of Contributors**  
Min Max

**Repository**  
Repository Name Repository Url

**Prerequisites**  
Maven Version

Select Dependency Type: Nothing

```
The packages found are: 2

package_name: Abbot Java GUI Test Library
package_version: 1.4.0
group_id: abbot
artifact_id: abbot
LicenseName: EPL
LicenseUrl: https://www.eclipse.org/legal/epl-v10.html
contributorName:
DeveloperName: Gerard Davisonr
DeveloperOrganization: Oralce
RepositoryName:
RepositoryUrl:
ContributorCount: 0
packageversiontableCount: 3

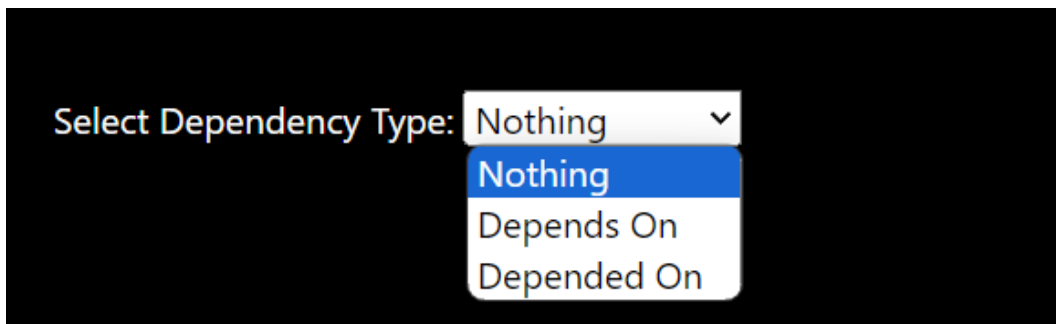
package_name: Abbot Java GUI Test Authoring Tool
package_version: 1.4.0
group_id: abbot
artifact_id: costello
LicenseName: EPL
LicenseUrl: https://www.eclipse.org/legal/epl-v10.html
contributorName:
DeveloperName: Gerard Davisonr
DeveloperOrganization: Oralce
RepositoryName:
RepositoryUrl:
ContributorCount: 0
packageversiontableCount: 1
```

### Σχήμα 5.1.2 Αναζήτηση πακέτων με περισσότερα από 1 πεδία

Τοποθετήθηκαν και στο *package-group\_id* και στο *DeveloperName* λέξεις και όταν έγινε η αναζήτηση τα αποτελέσματα δείχνουν ότι τα πακέτα περιέχουν στις πληροφορίες τους και τα δυο πεδία που αναφέρθηκαν.

### Αναζήτηση Εξαρτήσεων Πακέτου:

Ο χρήστης μπορεί να αναζητήσει τις εξαρτήσεις των πακέτων του με τη χρήση dropdown menu που επιτρέπει την επιλογή πακέτου και εμφανίζει δυναμικά τις εξαρτήσεις του, καθώς και τις συγκεκριμένες σχέσεις μεταξύ πακέτων. Μέσω των επιλογών "dependson" και "depended on", ο χρήστης μπορεί να προσδιορίσει ποια πακέτα θα εμφανίσει που εξαρτώνται από το επιλεγμένο πακέτο ή τα πακέτα που εξαρτώνται από αυτό.



**Σχήμα 5.1.3 Οι επιλογές που δίνονται για τις εξαρτήσεις πακέτων**

*Εδώ, ο χρήστης μπορεί να επιλέξει από το drop down menu τι θέλει να ψάξει. Αν θέλει κάποια εξάρτηση κάποιου πακέτου τότε παίρνει ή το “depends On” ή το “depended On” αλλιώς αν δεν θέλει να ψάξει κάποια εξάρτηση τότε διαλέγει την επιλογή “Nothing”.*

## Αναζήτηση Λέξεων σε Όνομα Πακέτου:

Επίσης, ο χρήστης μπορεί να αναζητήσει πακέτα βάση μιας συγκεκριμένης λέξης στο όνομα του πακέτου, αγνοώντας τη διάκριση πεζών-κεφαλαίων ή να εισάγει μια λέξη που περιέχει τον όρο που αναζητά. Με αυτό τον τρόπο το σύστημα θα συλλέξει όλα τα πακέτα που περιέχουν αυτή την λέξη.



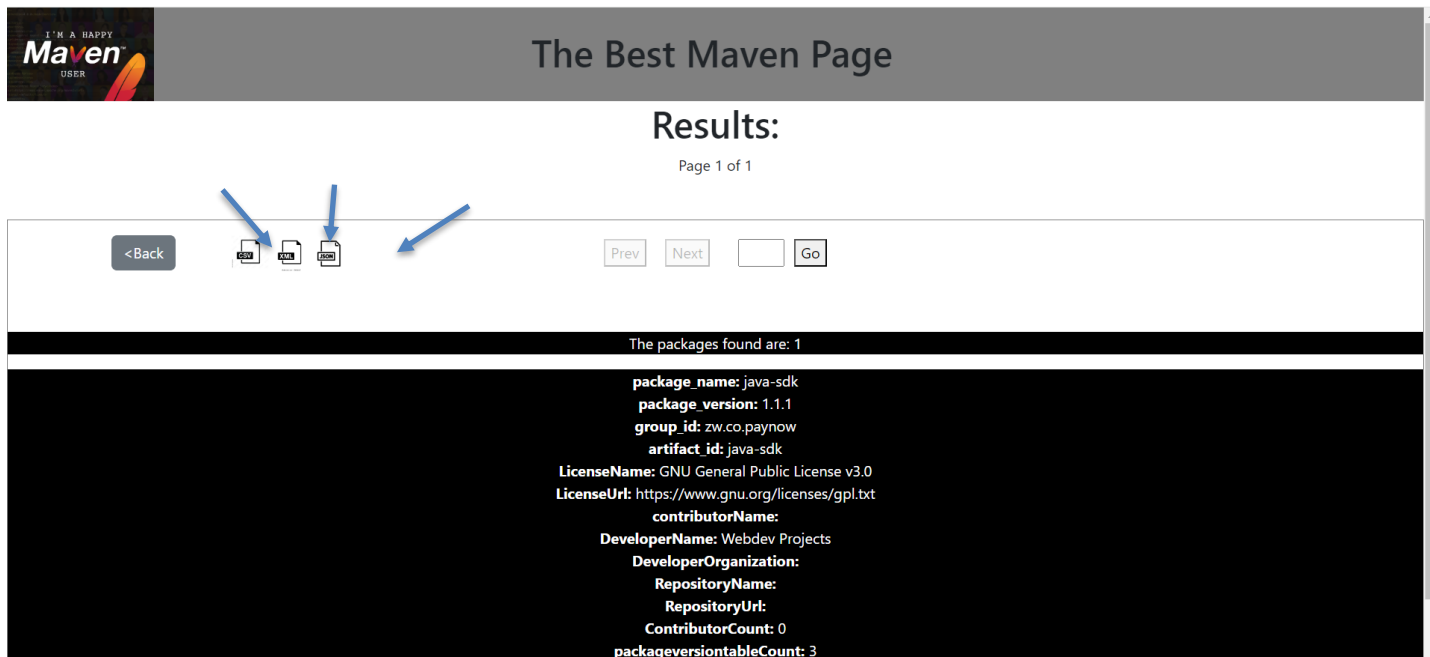
### Σχήμα 5.1.4 Παράδειγμα Αναζήτησης πακέτων που περιέχει την λέξη ‘ZW’

Στο πεδίο `group_id` τοποθετήθηκε η λέξη “ZW” και όταν πατήθηκε το κουμπί “search” αναγνώρισε το πακέτο το οποίο έχει και πεζά γράμματα αλλά και επίσης περιέχει την λέξη “ZW” μέσα της (`zw.co.paynow`).



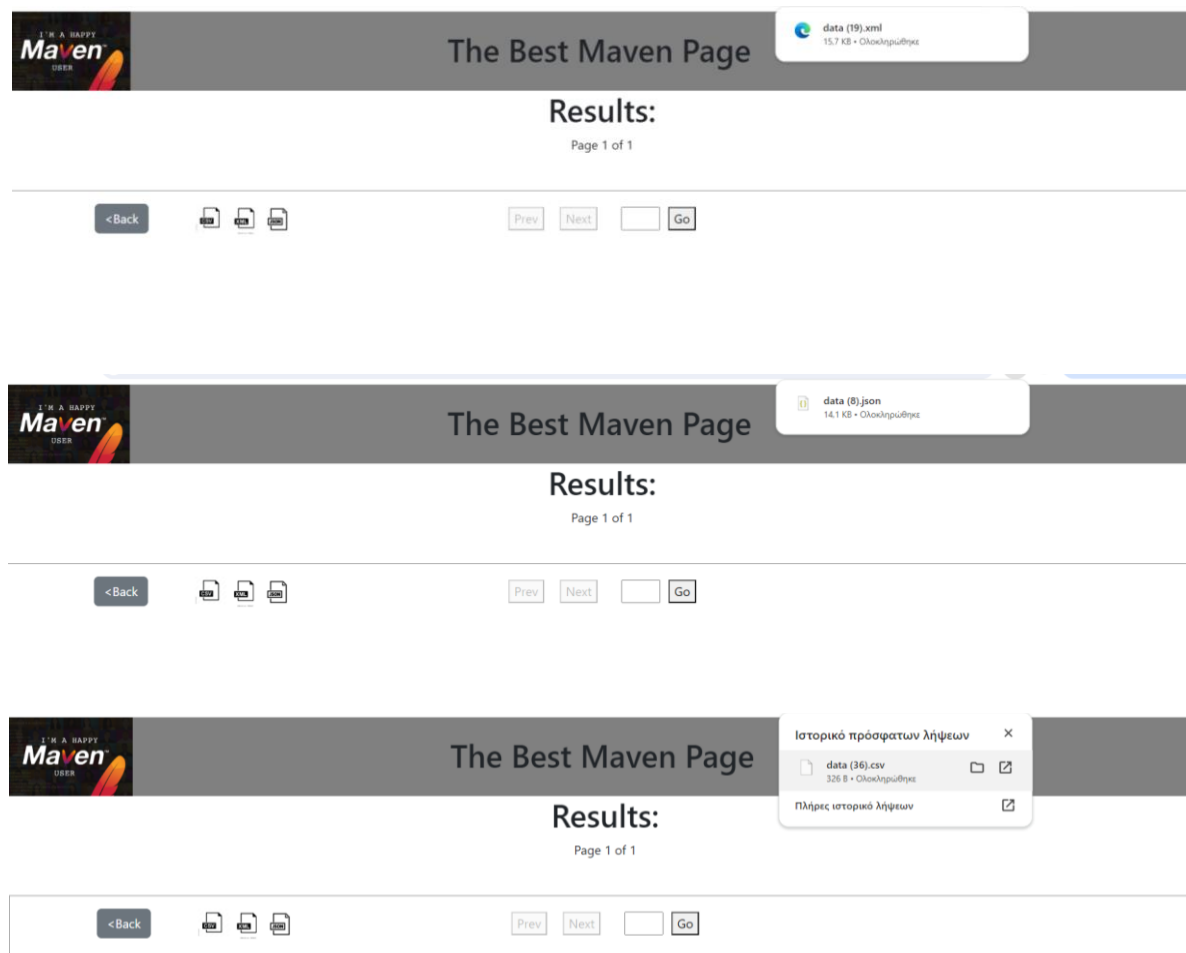
### Αποθήκευση σε διαφορετικά αρχεία:

Με την επιλογή αποθήκευσης των δεδομένων σε Csv, Json, και xml, ο χρήστης έχει ένα επιπλέον όφελος που θα τον βοηθήσει να αποθηκεύσει τα δεδομένα του όπως εκείνος τα χρειάζεται κάνοντας την εμπειρία του χρήστη ακόμη καλύτερη.



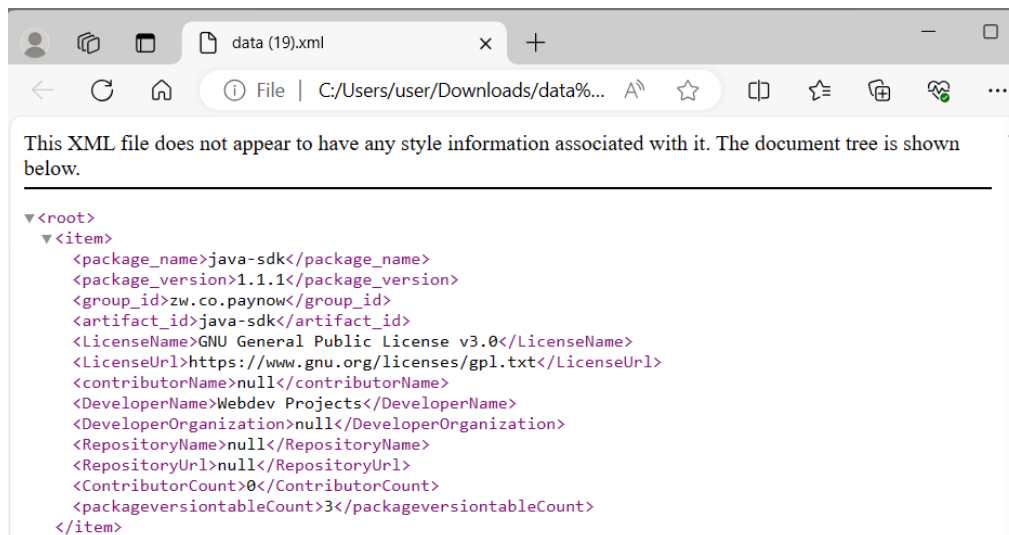
Σχήμα 5.1.5 Επιλογές αποθήκευσης δεδομένων

*Όπως βλέπετε και στα τόξα, εδώ υπάρχουν τα 3 είδη αποθήκευσης αρχείου όπου ο χρήστης μπορεί να επιλέξει σε ποιο από τα 3 θέλει να αποθηκεύσει τα αποτελέσματα του.*



**Σχήμα 5.1.6 Όταν ο χρήστης κάνει click σε ένα από τα 3 κουμπιά (XML, JSON, CSV).**

*Όταν ο χρήστης επιλέξει κάποιο από τα 3 αρχεία εμφανίζονται στην δεξιά πλευρά της σελίδας και μπορεί να τα ανοίξει.*



**Σχήμα 5.1.7 Αρχείο XML**



**Σχήμα 5.1.8 Αρχείο JSON**

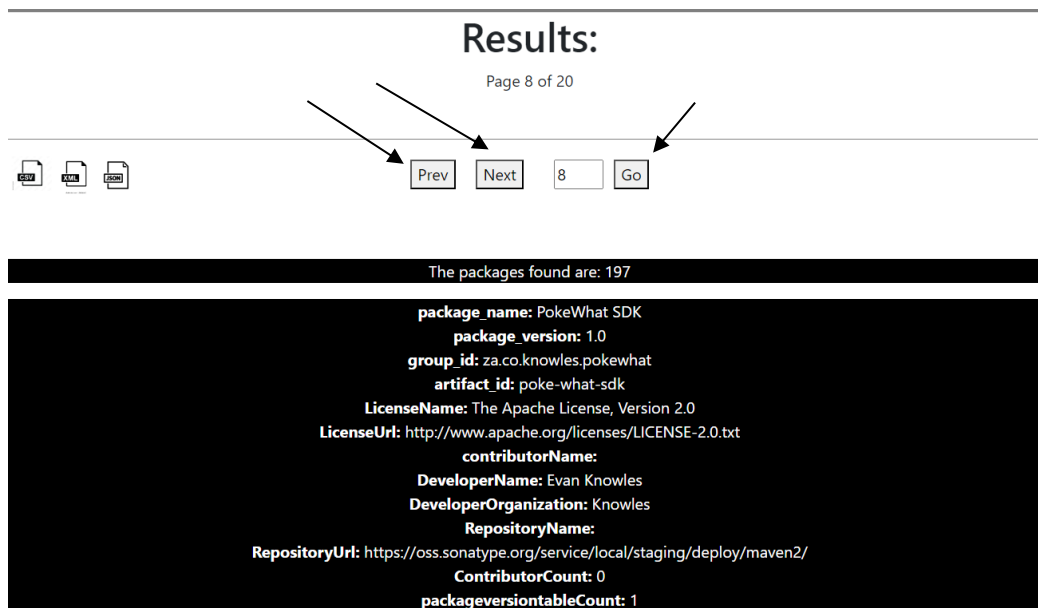


**Σχήμα 5.1.9 Αρχείο CSV**

*Παρουσιάζεται η δομή του κάθε αρχείου ξεχωριστά*

## Μετάβαση στην επόμενη/προηγούμενη σελίδα:

Επιλέγοντας τον επιθυμητό αριθμό σελίδας, ο χρήστης μπορεί να μεταβεί σε οποιαδήποτε σελίδα χρειάζεται ανάλογα με τα δεδομένα ή μπορεί να αναζητήσει μια συγκεκριμένη σελίδα τοποθετώντας τον αριθμό σελίδας και κάνοντας κλικ στην αναζήτηση “Go”.



### Σχήμα 5.1.10 Μετάβαση στην επόμενη/προηγούμενη σελίδα ή με την επιλογή ‘Go’

*Ο χρήστης μπορεί να κάνει click στο κουμπί “Prev” για να πάει στην προηγούμενη σελίδα, “Next” για να πάει στην επόμενη σελίδα καθώς και να πληκτρολογήσει τον αριθμό της σελίδας στο πεδίο που υπάρχει και να πατήσει το κουμπί “Go” για να μεταφερθεί στη σελίδα που αναζητά.*

Με αυτές τις λειτουργίες, παρέχουμε στον χρήστη ένα πλήρες και ευέλικτο εργαλείο για τη διαχείριση των εξαρτήσεων και των πακέτων του στο Maven Central Repository.

# Κεφάλαιο 6

## Αξιολόγηση

---

|                                                   |    |
|---------------------------------------------------|----|
| 6.1 MVN Repository                                | 38 |
| 6.2 Σύγκριση                                      | 39 |
| 6.2.1 Πλεονεκτήματα Συστήματος σε σχέση με το MVN | 39 |
| 6.2.2 Μειονεκτήματα Συστήματος σε σχέση με το MVN | 39 |
| 6.3 Αξιολόγηση συστήματος                         | 40 |

---

### 6.1 MVN Repository

Μέσω του Maven Repository, οι προγραμματιστές μπορούν να περιηγηθούν σε μια λεπτομερή συλλογή από βιβλιοθήκες και εξαρτήσεις λογισμικού που φιλοξενούνται στο Maven Central Repository [74]. Επίσης, παρέχει διάφορα εργαλεία αναζήτησης, φιλτραρίσματος και πλοήγησης, καθώς και πληροφορίες για κάθε βιβλιοθήκη, συμπεριλαμβανομένων εκδόσεων, εξαρτήσεων και άλλων λεπτομερειών. Το Maven Repository επιτρέπει στους προγραμματιστές να εντοπίζουν εύκολα τις εξαρτήσεις που χρειάζονται για τα έργα τους, να διαχειρίζονται τις εκδόσεις και τις συμβατικότητες, και να βρίσκουν πληροφορίες σχετικά με τις άδειες χρήσης και τις πλατφόρμες υποστήριξης. Συνολικά, λειτουργεί ως πολύτιμο εργαλείο για την ανάπτυξη λογισμικού με τη χρήση του Apache Maven.

## 6.2 Σύγκριση

### 6.2.1 Πλεονεκτήματα Συστήματος σε σχέση με το MVN

Τα πλεονεκτήματα του συστήματός σε σχέση με το Maven Repository είναι πολλά και σημαντικά, βοηθώντας τον χρήστη σε μεγάλο βαθμό στη μελέτη του και στις ανάγκες του. Καταρχάς, το σύστημά παρέχει τη δυνατότητα εξαγωγής δεδομένων σε μορφή CSV, XML και JSON, δίνοντας στον χρήστη τη δυνατότητα να αποθηκεύσει τα δεδομένα του σε μορφή που είναι κατανοητή για εκείνον. Αντίθετα, για να εξάγει δεδομένα σε μορφή CSV από το Maven Repository, απαιτείται η δημιουργία ενός προτύπου πακέτου Java (JAR) που να περιλαμβάνει το αποτέλεσμα της αναζήτησης σε CSV [75], μια διαδικασία που είναι πιο πολύπλοκη και χρονοβόρα για τον χρήστη

Επιπλέον, το σύστημά παρέχει τις επιλογές "dependsOn" και "dependedOn", κάτι που δεν υπάρχει αυτόματα στο Maven Repository. Ο χρήστης μπορεί να χρησιμοποιήσει αυτές τις επιλογές για να ψάξει τις εξαρτήσεις των πακέτων, κάτι που διευκολύνει τη μελέτη του. Στο Maven Repository, αυτή η δυνατότητα δεν είναι διαθέσιμη ενσωματωμένα, κάτι που σημαίνει ότι ο χρήστης θα πρέπει να βασίζεται σε άλλα εργαλεία ή πηγές πληροφοριών για να διερευνήσει [76] τις εξαρτήσεις μεταξύ των πακέτων. Τέλος, το σύστημά μας παρέχει άμεση πρόσβαση σε όλες τις απαραίτητες πληροφορίες για τα πακέτα κατά την αναζήτηση. Αυτό καθιστά τη διαδικασία πιο εύχρηστη σε σύγκριση με το Maven Repository, όπου η αναζήτηση και η ανάλυση των πληροφοριών απαιτούν περισσότερο χρόνο και προσπάθεια από τον χρήστη.

### 6.2.2 Μειονεκτήματα Συστήματος σε σχέση με το MVN

Αρχικά, το Maven Repository (MVN) παρέχει περισσότερες λεπτομέρειες σε σύγκριση με το σύστημα. Περιλαμβάνει κατηγορίες, που επιτρέπουν στους χρήστες να αναζητήσουν εύκολα ό,τι χρειάζονται. Επιπλέον, παρέχει δυνατότητα επικοινωνίας με την ομάδα υποστήριξης της ιστοσελίδας [76], έτσι ώστε να λυθούν προβλήματα που μπορεί να προκύψουν, και την επιλογή «popular» που αφορά τα δημοφιλή εξαρτήματα λογισμικού που χρησιμοποιούνται ευρέως από την κοινότητα των προγραμματιστών. Αυτά τα εξαρτήματα έχουν συχνά υψηλό αριθμό λήψεων και είναι διαδεδομένα σε πολλά

έργα λογισμικού. Η επιλογή "Popular" επιτρέπει στους χρήστες να βρίσκουν γρήγορα και εύκολα τα πιο δημοφιλή εξαρτήματα που χρησιμοποιούνται στην κοινότητα [77]. Επίσης, ακόμη ένα μειονέκτημα του συστήματος σε σχέση με το Maven Repository, είναι η ένδειξη "Indexed Repositories (1997+)" που υποδεικνύει τον συνολικό αριθμό των αποθετηρίων (repositories) που έχουν εντοπιστεί και ευρετηριαστεί από το σύστημα αναζήτησης του Maven. Αυτό σημαίνει ότι το Maven Repository έχει εντοπίσει και επεξεργαστεί περισσότερα από 1997 αποθετήρια[7] από όλο το φάσμα του λογισμικού του οικοσυστήματος Maven. Αυτό καθιστά δυνατή την πρόσβαση των προγραμματιστών σε μια ευρεία γκάμα πακέτων και εξαρτημάτων λογισμικού που είναι διαθέσιμα μέσω του Maven Repository.

Τέλος, ένα ακόμη εξίσου σημαντικό μειονέκτημα που έχει το σύστημα μας σε σχέση με το Maven Repository είναι οι "Δημοφιλείς Ετικέτες" (Popular Tags) [78] όπου αναφέρονται στις ετικέτες που χρησιμοποιούνται συχνά για την κατηγοριοποίηση των αποθετηρίων και των πακέτων λογισμικού στην πλατφόρμα. Οι δημοφιλείς ετικέτες παρέχουν μια επιπλέον προοπτική για τους χρήστες να αναζητήσουν και να περιηγηθούν σε πακέτα λογισμικού που ανήκουν σε συγκεκριμένες κατηγορίες ή τεχνολογίες. Οι ετικέτες μπορεί να περιλαμβάνουν ονόματα γλωσσών προγραμματισμού, πλατφορμών, βιβλιοθηκών, εργαλείων και άλλων σχετικών χαρακτηριστικών που βοηθούν στον προσδιορισμό του περιεχομένου και της χρήσης των πακέτων. Φυσικά, είναι λογικό το Maven Repository να έχει περισσότερες λεπτομέρειες από το σύστημα μας καθώς χρησιμοποιείται ευρέως από πολλούς προγραμματιστές και έχει αναπτυχθεί και βελτιωθεί κατά τη διάρκεια πολλών ετών. Η εκτενής χρήση του από πολλούς ανθρώπους και η μακρά ιστορία του αποτελούν ένδειξη της αξιοπιστίας και της χρησιμότητάς του.

### **6.3 Αξιολόγηση συστήματος**

Αφού τελείωσα το σύστημα μου το έχω δείξει σε πολλούς φίλους μου για να το δούν και να μου πούν τα σχόλια που έχουν για αυτό. Όλοι σχεδόν μου είπαν πως είναι πολύ καλό και ειδικά για ένα χρόνο δουλείας δούλεψε εξαιρετικά. Ένας φίλος μου αφού του το έδειξα μου είπε ένα πρόβλημα που υπήρχε και έπρεπε να το διορθώσω επειδή ήταν σημαντικό. Αυτό το πρόβλημα αναφέρεται στο ότι αν ο χρήστης δεν τοποθετήσει κάτι σε κάποιο πεδίο αυτό τρέχει και δεν εμφανίζει τίποτα που αυτό ήταν πρόβλημα και μου

έδωσε σαν λύση να βάλω κάποιο περιορισμό, να εμφανίζω κάποιο μήνυμα έτσι ώστε αν ο χρήστης δεν τοποθετήσει κάτι να βγάζει το κατάλληλο alert. Και έτσι και έκανα, Έχω τοποθετήσει ένα window.alert μήνυμα που επιστρέφει στον χρήστη πως δεν έχετε βάλει κάτι στο πεδίο ώστε να τον προειδοποιήσει πως αυτό που κάνει δεν είναι επιτρεπτό. Πιστεύω με αυτή την λύση το πρόβλημα μου έχει λυθεί σωστά και με αποδοτικό τρόπο.



# Κεφάλαιο 7

## Συμπεράσματα

---

|                        |    |
|------------------------|----|
| 7.1 Επίλογος           | 42 |
| 7.2 Μελλοντική Δουλειά | 43 |

---

### 7.1 Επίλογος

Συνοψίζοντας, τα οικοσυστήματα αποτελούν κρίσιμο παράγοντα στον τομέα της επιστήμης και της τεχνολογίας, συμβάλλοντας σημαντικά στην πρόοδο και την ανάπτυξη. Η εύκολη πρόσβαση σε πληροφορίες και πακέτα, η βελτίωση της διαχείρισης εξαρτήσεων καθώς και η ευκολία χρήσης λογισμικών είναι από τους σημαντικότερους παράγοντες των οικοσυστημάτων. Μέσω της δημιουργίας μιας πλατφόρμας, η οποία επιτρέπει στους χρήστες να εντοπίζουν και να αναζητούν πακέτα με μεγάλη ευκολία παρέχουμε μια ευχάριστη εμπειρία στους χρήστες καθώς μπορούν να δουν και να μελετήσουν το κάθε πακέτο όποτε αυτοί επιθυμούν. Με αυτόν τον τρόπο, η προσέγγισή μας εξυπηρετεί τις ανάγκες των χρηστών της Maven και ταυτόχρονα παρέχει ένα περιβάλλον που ενθαρρύνει την ανάπτυξη και την καινοτομία. Οι χρήστες μας έχουν τη δυνατότητα να αναζητούν και να επιλέγουν πακέτα με βάση τις ανάγκες τους, ενώ ταυτόχρονα έχουν πρόσβαση σε πλήρεις και αξιόπιστες πληροφορίες για κάθε πακέτο. Επιπλέον, η διαχείριση των εξαρτήσεων γίνεται ευκολότερη και πιο αποτελεσματική, κάτι που συμβάλλει στη συνολική αποτελεσματικότητα της διαδικασίας ανάπτυξης λογισμικού. Εν κατακλείδι, η συνεχής εξέλιξη και βελτίωση του συστήματός αποτελεί συμβολή στην πρόοδο της τεχνολογίας και της κοινότητας των προγραμματιστών. Με αυτόν τον τρόπο, διασφαλίζουμε ότι οι χρήστες θα είναι ευχαριστημένοι, προωθώντας έτσι και την καινοτομία και την ανάπτυξη στον τομέα της τεχνολογίας λογισμικού.

## 7.2 Μελλοντική Δουλειά

Για ένα σύστημα όπως αυτό, η προσπάθεια ενός χρόνου είναι αναμφισβήτητα ανεπαρκής και υπάρχουν ακόμη ορισμένα πράγματα που μπορούν να αλλάξουν για να βελτιωθεί και να αυξηθεί η αποτελεσματικότητα του χρήστη. Αρχικά, η πρώτη σημαντική αλλαγή που μπορεί να γίνει στο σύστημα μας είναι η σελίδα Login/Sign - In όπου κάθε χρήστης θα μπορεί να βλέπει τα δικά του πακέτα όπως για παράδειγμα τα αγαπημένα του, το ιστορικό αναζήτησης για πακέτα που είχε αναζητήσει στο παρελθόν καθώς και λέξεις κλειδιά που έχει αναζητήσει ο ίδιος στην μελέτη του. Με αυτό τον τρόπο, σε περίπτωση που ο χρήστης ξεχάσει ένα όνομα ή κάποιο group\_id να μπορεί να το εντοπίσει γρήγορα χωρίς να χρειάζεται να θυμάται ποιο είναι.

Επίσης, κάτι άλλο που μπορεί να προστεθεί στο σύστημα ώστε να βελτιωθεί η χρηστικότητα του, είναι να υπάρχει ένα menu με τα πιο γνωστά πακέτα και βιβλιοθήκες που αυτό μπορεί να είναι ιδιαίτερα χρήσιμο για κάποιο χρήστη που ψάχνει για τα πιο αναγνωρίσιμα και γνωστά πακέτα. Ακόμη κάτι που θα έκανε το σύστημα πιο αποδοτικό είναι να προστεθεί κάποιο είδος chat ώστε ο χρήστης αν δεν θυμάται κάποιο πακέτο ή κάποια εξάρτηση να μπορεί απλά να στείλει ένα μήνυμα στο chat και να του απαντήσει αμέσως με το διαθέσιμο όνομα του πακέτου που ψάχνει. Αυτά θα ήταν πολύ αποδοτικές αλλαγές και θα έκαναν την σελίδα να εκτοξευθεί.

# Βιβλιογραφία

[1] Adducul, R. (2020). Crowdsourcing Technology for Classroom Learning. [Online]. [https://www.researchgate.net/profile/RamonsitoAdducul/publication/344856838\\_Crowdsourcing\\_Technology\\_for\\_Classroom\\_Learning/links/64d3b357d3e680065aa3b4f3/Crowdsourcing-Technology-for-Classroom-Learning.pdf](https://www.researchgate.net/profile/RamonsitoAdducul/publication/344856838_Crowdsourcing_Technology_for_Classroom_Learning/links/64d3b357d3e680065aa3b4f3/Crowdsourcing-Technology-for-Classroom-Learning.pdf)

[2] Jansen, S., Brinkkemper, S., & Souer, J. (2009). From Integration to Composition: On the impact of software product lines, global development, and ecosystems. In Proceedings of the 2009 IEEE 31<sup>st</sup> International Conference on Software Engineering- Companion Volume (pp. 391-392). [https://www.researchgate.net/publication/220376098\\_From\\_Integration\\_to\\_Composition\\_On\\_the\\_impact\\_of\\_software\\_product\\_lines\\_global\\_development\\_and\\_ecosystems](https://www.researchgate.net/publication/220376098_From_Integration_to_Composition_On_the_impact_of_software_product_lines_global_development_and_ecosystems)

[3] Brush, K. (2019). What is a Digital Ecosystem? [online] SearchCIO. <https://www.techtarget.com/searchcio/definition/digital-ecosystem>

[4,12,22] Zhao, L., Chen, S., Xu, Z., Liu, C., Zhang, L., Wu, J., Sun, J. and Liu, Y. (n.d.). Software Composition Analysis for Vulnerability Detection: An Empirical Study on Java Projects. [online] [https://dl.acm.org/doi/pdf/10.1145/3611643.3616299?casa\\_token=5SkLLRTwuSIAAA:AA:zm5abMvNcwyngRQgQUGCERUy0YWrX989uq05KNrbzY1nyp6PX-zCgC\\_Ea48NRHo0qpslJKqTbZY](https://dl.acm.org/doi/pdf/10.1145/3611643.3616299?casa_token=5SkLLRTwuSIAAA:AA:zm5abMvNcwyngRQgQUGCERUy0YWrX989uq05KNrbzY1nyp6PX-zCgC_Ea48NRHo0qpslJKqTbZY)

[5] Boudi, A., Farris, I., Bagaa, M. and Talbe, T. (n.d.). Assessing Lightweight Virtualization for Security-as-a-Service at the Network Edge. [online] <https://arxiv.org/pdf/2211.11197>

[6] Smu.edu.sg. (2024). [https://ink.library.smu.edu.sg/cgi/viewcontent.cgi?article=9174&context=sis\\_research](https://ink.library.smu.edu.sg/cgi/viewcontent.cgi?article=9174&context=sis_research)

[7] MVN Repository

[Maven Repository: Repositories \(mvnrepository.com\)](https://mvnrepository.com)

[8] Yang, T., & Cui, L. (2021). Data augmentation with generative adversarial networks for improving detection performance of remote sensing images. Pattern Recognition Letters, 149, 1-9.

<https://www.sciencedirect.com/science/article/pii/S0164121221002314>

[9] Apache.org. (2013). Maven – Introduction. [online]

<https://maven.apache.org/what-is-maven.html>

[10] Porter Trygve, B. (2013). Maven – Introduction to the Dependency Mechanism. [online] Apache.org.

<https://maven.apache.org/guides/introduction/introduction-to-dependency-mechanism.html>

[11] The AutoBot (2024). Mastering Maven: Dependency Management and Build Automation for JavaProjects. [online] Medium.

<https://medium.com/@theautobot/mastering-maven-dependency-management-and-build-automation-for-java-projects-f3f61c7bc86d>

[13] Arcelli Fontana, F., Di Pompeo, D., Ferme V., Spinelli, S., & Tamburri, D. A. (2016). Defining Metrics for Microservices Architecture. In Proceedings of the 2016 IEEE/ACM 13<sup>th</sup> Working Conference on Mining Software Repositories (MSR) (PP. 1-4).

[https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7515498&casa\\_token=hdcUH4Hrf\\_sAAAAA:OPGh\\_DI50kWGwPVccUKeiSk-f4DMIbmuWmmKlBAaGGMoui\\_jG-PJDG7cj3oiIaB\\_dXME9t&tag=1](https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7515498&casa_token=hdcUH4Hrf_sAAAAA:OPGh_DI50kWGwPVccUKeiSk-f4DMIbmuWmmKlBAaGGMoui_jG-PJDG7cj3oiIaB_dXME9t&tag=1)

[14] Amos, L.S. and Magadza, E.T. (2024). 5V's of Big Data Shifted to Suite the Context of Software Code: Big Code for Big Software Projects. *Journal of Artificial Intelligence and Big Data*, [online] pp.36-47.

<https://www.scipublications.com/journal/index.php/jaibd/article/view/911>

[15] İşcan, H., & Turetken, O. (2014). Service Design Studio (SDS) – The execution environment for the Service Design Studio

[https://www.researchgate.net/profile/Huelya\\_Iscan/publication/262060254\\_Service\\_Design\\_Studio\\_SDS\\_The\\_execution\\_environment\\_for\\_the\\_Service\\_Design\\_Studio/links/00b495368b6e2d0438000000.pdf](https://www.researchgate.net/profile/Huelya_Iscan/publication/262060254_Service_Design_Studio_SDS_The_execution_environment_for_the_Service_Design_Studio/links/00b495368b6e2d0438000000.pdf)

[16] maven.apache.org. (n.d.). Maven – Introduction to the POM. [online]

<https://maven.apache.org/guides/introduction/introduction-to-the-pom.html#:~:text=What%20is%20a%20POM%3F,Maven%20to%20build%20the%20project.>

[17] maven.apache.org. (n.d.). Maven – POM Reference. [online]

<https://maven.apache.org/pom.html>

[18] Porter Trygve, B. (2013). Maven – Introduction to the Dependency Mechanism. [online] Apache.org.

<https://maven.apache.org/guides/introduction/introduction-to-dependency-mechanism.html>

[19] Maven.apache.org. (n.d.). Maven – Introduction to the Build Lifecycle. [online]

<https://maven.apache.org/guides/introduction/introduction-to-the-lifecycle.html>

[20] Z. Wei, W. Li, X. Liu, and Y. Zhang, "A Deep Learning Approach for Real-Time Traffic Prediction Using Long Short-Term Memory Neural Network," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 4, pp. 1234-1243, April 2024.

<https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=10235318>

[21] N. Patel and A. Agrawal, "IoT-Based Smart Healthcare Kit," in *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8817-8824, Oct. 2019.

[https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8816809&casa\\_token=XD8yX5\\_zwgEAAAAA:zohJ9Sus31a3HXFUUuyJqOaHpbup7\\_dnOebY1APiUp2K9PEEHwmbcY4z0gjL03PU-J68\\_W](https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8816809&casa_token=XD8yX5_zwgEAAAAA:zohJ9Sus31a3HXFUUuyJqOaHpbup7_dnOebY1APiUp2K9PEEHwmbcY4z0gjL03PU-J68_W)

[23] Noh, Y. (2015). Imaging Library 4.0: Creating a Model for Future Libraries. The Journal of Academic Librarianship, [online] 41(6), pp.786-797.

<https://www.sciencedirect.com/science/article/pii/S0099133315001780?via%3Dihub>

[24] Randhawa and Sukhwinder (n.d.). Open Source Software and Libraries Item Type Conference Paper. [online]

<https://repository.arizona.edu/bitstream/handle/10150/105743/?sequence=1>

[25] Tariq Rahim Soomro and Mohammad Hasan Saeed Al-alwi, "Achieving Software Release Management and Continuous Integration using Maven, Jenkins, and Artifactory," May 2024.

[https://www.researchgate.net/profile/TariqSoomro2/publication/330008070\\_Achieving\\_Software\\_Release\\_Management\\_and\\_Continuous\\_Integration\\_using\\_Maven\\_Jenkins\\_and\\_Artifactory/links/5c377a9d299bf12be3bcee58/Achieving-Software-Release-Management-and-Continuous-Integration-using-Maven-Jenkins-and-Artifactory.pdf](https://www.researchgate.net/profile/TariqSoomro2/publication/330008070_Achieving_Software_Release_Management_and_Continuous_Integration_using_Maven_Jenkins_and_Artifactory/links/5c377a9d299bf12be3bcee58/Achieving-Software-Release-Management-and-Continuous-Integration-using-Maven-Jenkins-and-Artifactory.pdf)

[26] Paschechenko, I., Vu, D.-L. and Massacci, F. (2020). A Qualitative Study of Dependency Management and its Security Implications. Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security.

[https://dl.acm.org/doi/pdf/10.1145/3372297.3417232?casa\\_token=ndiEBdnYkkYAAA\\_AA:nXEVZ9jOPEnKlkM7BVY2dgSUaEWly9n2yMG5l70a9oDkMn6hrNQjMUvuDTd04sMGcTG7gXjfHk](https://dl.acm.org/doi/pdf/10.1145/3372297.3417232?casa_token=ndiEBdnYkkYAAA_AA:nXEVZ9jOPEnKlkM7BVY2dgSUaEWly9n2yMG5l70a9oDkMn6hrNQjMUvuDTd04sMGcTG7gXjfHk)

[27] Dzone.com. (n.d.). Dependency Management and Versioning with a Maven Multi-Module Project - DZone. [online]

<https://dzone.com/articles/maven-multi-module-project-with-versioning>

[28] Barnes, S.J. and Pressey, A.D. (2012). In Search of the ‘Meta-Maven’: An Examination of Market Maven Behavior across Real-Life, Web, and Virtual World Marketing Channels. *Psychology and Marketing*, 29(3), pp.167-185.

[https://onlinelibrary.wiley.com/doi/pdfdirect/10.1002/mar.20513?casa\\_token=SEGTmk9WiXUAAAAA:caa94sCYvnE3sLAj95hTBik6hUS5PgLEliQivzDAu8IXZNF879eIUfjvqPZ9ezNARnPW4pFVEIV4Q](https://onlinelibrary.wiley.com/doi/pdfdirect/10.1002/mar.20513?casa_token=SEGTmk9WiXUAAAAA:caa94sCYvnE3sLAj95hTBik6hUS5PgLEliQivzDAu8IXZNF879eIUfjvqPZ9ezNARnPW4pFVEIV4Q)

[29] Chung, L. and do Prado Leite, J.C.S. (2009). On Non-Functional Requirements in Software Engineering. *Conceptual Modeling: Foundations and Applications* pp.363-379.

[https://edisciplinas.usp.br/pluginfile.php/5281085/mod\\_resource/content/1/On\\_Non-Functional\\_Requirements\\_in\\_Software\\_Engineer.pdf](https://edisciplinas.usp.br/pluginfile.php/5281085/mod_resource/content/1/On_Non-Functional_Requirements_in_Software_Engineer.pdf)

[30] Loup Meurice, Nagy, C. and Cleve, A. (2016). Static Analysis of Dynamic Database Usage in Java Systems. *Lecture notes in computer science*, pp.491-506.

[https://link.springer.com/chapter/10.1007/978-3-319-39696-5\\_30](https://link.springer.com/chapter/10.1007/978-3-319-39696-5_30)

[31] Ouni, A., Kula, R.G., Kessentini, M., Ishio, T., German, D.M. and Inoue, K. (2017). Search-based software library recommendation using multi-objective optimization. *Information and Software Technology*, 83, pp.55-75.

[https://www.sciencedirect.com/science/article/pii/S0950584916303652?casa\\_token=xEcq\\_i7n9pMAAAAA:7MzDJdb0SxQTJcjo7ys2ZJMzEXut5mfirkDu0vIhfxFWwwmf2cPv52ymlFFeg\\_xVbpwQhV](https://www.sciencedirect.com/science/article/pii/S0950584916303652?casa_token=xEcq_i7n9pMAAAAA:7MzDJdb0SxQTJcjo7ys2ZJMzEXut5mfirkDu0vIhfxFWwwmf2cPv52ymlFFeg_xVbpwQhV)

[32] Osma, A.M.S. (2019). A novel big data analytics framework for smart cities. *Future Generation Computer Systems*, 91(1), pp.620-633.

[https://www.sciencedirect.com/science/article/pii/S0167739X17307446?casa\\_token=RtZaxsz6vyQAAAAA:t\\_czhAARFmsceJ8GqpQ0Wn\\_igjtreCEX6foiVBI8rtnaf3EWpIuGGBUJmslo3T4UZfwHQ2V](https://www.sciencedirect.com/science/article/pii/S0167739X17307446?casa_token=RtZaxsz6vyQAAAAA:t_czhAARFmsceJ8GqpQ0Wn_igjtreCEX6foiVBI8rtnaf3EWpIuGGBUJmslo3T4UZfwHQ2V)

[33] Y. Wang, G. Shen, L. Chen, X. Li, and L. Hu, "Feature Fusion and Attention Mechanism for Anomaly Detection in Time Series Data," in *IEEE Access*, vol. 8, pp. 37705-37716, 2020.

<https://ieeexplore.ieee.org/abstract/document/10061380>

[34,37] Srivastava, J., Cooley, R., Deshpande, M. and Tan, P.-N (2000). Web usage mining. ACM SIGKDD Explorations Newsletter, 1(2), p.12.

<https://dl.acm.org/doi/pdf/10.1145/846183.846188>

[35] Altexsoft (2022). Non-functional Requirements: Examples, Types, How to Approach. [online] AltexSoft.

<https://www.altexsoft.com/blog/non-functional-requirements/>

[36] Mahmoud, A., and Williams, G. (2016). Detecting, classifying, and tracing non-functional software requirements. Requirements Engineering, 21(3), pp.357-381.

<https://link.springer.com/article/10.1007/s00766-016-0252-8>

[38] Vsupalov.com. (2019) How Does the Frontend Communicate with the Backend? [online]

<https://vsupalov.com/how-backend-and-frontend-communicate/>

[39] Fariz, M., Lazuardy, S. and Anggrainin, D. (2022). Modern Front End Web Architectures with React.Js and Next.Js. International Research Journal of Advanced Engineering and Science, [online] 7(1), pp.132- 141.

<https://irjaes.com/wp-content/uploads/2022/02/IRJAES-V7N1P162Y22.pdf>

[40] Patel, D. (2023). CRUD operations in Spring Boot and MySQL (Rest APIs). [online] Medium.

<https://deeppatel23.medium.com/crud-operations-in-spring-boot-and-mysql-rest-apis-9ca0c15bdc73>



- [41] Sumo Logic. (n.d.). Discover what crud is | definition and overview. [online]  
<https://www.sumologic.com/glossary/crud/#:~:text=CRUD%20is%20an%20acronym%20from,%2C%20read%2C%20update%20and%20delete.>
- [42] Help.natureserve.org. (n.d.). How to Read a Relational Data Model. [online]  
[https://help.natureserve.org/biotics/content/DataModel/DM\\_HowToRead\\_DataModel.htm#:~:text=Primary%20key%20\(PK\)%20%2D%20value,inherited%20from%20some%20other%20table.](https://help.natureserve.org/biotics/content/DataModel/DM_HowToRead_DataModel.htm#:~:text=Primary%20key%20(PK)%20%2D%20value,inherited%20from%20some%20other%20table.)
- [43] Wittern, E., Suter, P. and Pajagopalan, S. (2016). A look at the dynamics of the JavaScript package ecosystem. Proceedings of the 13<sup>th</sup> International Workshop on Mining Software Repositories – MSR '16.  
[https://dl.acm.org/doi/pdf/10.1145/2901739.2901743?casa\\_token=AnpirB1XI40AAAAA:A:TnqW5Yn2LAQlZ2w4HkssYXGwxNLNEpMCoXM4iPPxayGVGwbHe2OB-EEiscv2VL\\_JkCMQGdY5MH0h](https://dl.acm.org/doi/pdf/10.1145/2901739.2901743?casa_token=AnpirB1XI40AAAAA:A:TnqW5Yn2LAQlZ2w4HkssYXGwxNLNEpMCoXM4iPPxayGVGwbHe2OB-EEiscv2VL_JkCMQGdY5MH0h)
- [44] Noel (2024). How To Go About Using JDBC To Connect A Java Application To Any Database. [online] Medium.  
<https://medium.com/@noel.B/how-to-go-about-using-jdbc-to-connect-a-java-application-to-any-database-49d05e13c2a6>
- [45] N. Hovakimyan, E. N. Johnson, P. P. Khargonekar, S. J. Yurkovich and V. Dobrokhodov, "An approach to quadrotor control," in International Journal of Robust and Nonlinear Control, vol. 18, no. 10, pp. 1038-1052, 2008  
<https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=33da487c929a5286413605e111c250e1ac075e28>
- [46] Lawson, B., Sharp, R. and Lauke, P. (n.d.). Introducing HTML5. [online]  
<https://ptgmedia.pearsoncmg.com/images/9780321687296/samplepages/0321687299.pdf>
- [47] HowStuffWorks. (2011). How HTML5 Works. [online] Available at:  
<https://computer.howstuffworks.com/html-five.htm>

[48] Taraldsvik, M. (2011). Exploring the Future: is HTML5 the solution for GIS Applications on the World Wide Web? [online]

<https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=dcbb0e136fb693ee06597aa44e13af876ce4e7fa>

[49] Mozilla (2019). CSS: Cascading Style Sheets. [online] MDN Web Docs.

<https://developer.mozilla.org/en-US/docs/Web/CSS>

[50] Olumide, S. (2023). What is Java Used for in 2023? The Java Programming Language and Java Platform Strengths. [online] freeCodeCamp.org.

<https://www.freecodecamp.org/news/what-is-java-used-for/>

[51] Morris, S. (2016). What Exactly is Visual Design, Anyway? [online] Skillcrush.

<https://skillcrush.com/blog/whatisbootstrap/#:~:text=Bootstrap%20provides%20develop%20with%20a,common%20UI%20elements%20from%20scratch.>

[52] DEV Community. (2020). Benefits of Bootstrap framework with it's example. [online]

[https://dev.to/brijesh\\_dobariya/benefits-of-bootstrap-framework-with-it-s-example-2i26](https://dev.to/brijesh_dobariya/benefits-of-bootstrap-framework-with-it-s-example-2i26)

[53] Coursera (2023). What Does a Back-End Developer Do? [online] Coursera.

<https://www.coursera.org/articles/back-end-developer>

[54] Oracle (2019). MySQL: MySQL 8.0 Reference Manual: 1.3.1 What is MySQL? [online] Mysql.com.

<https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>

[55] S. Chen, H. Xu, and Y. Li, "Edge Computing in IoT-Based Manufacturing," *Future Generation Computer Systems*, vol. 97, pp. 395-403, Aug. 2019.

<https://www.sciencedirect.com/science/article/pii/S0167739X18308732?via%3Dihub>

[56] Parikshith Reddy Baddam, "Unlocking PHP's Potential: An All-Inclusive Approach to Server-Side Scripting," May 2024.

[https://www.researchgate.net/profile/ParikshithReddyBaddam/publication/377025333\\_Unlocking\\_PHP's\\_Potential\\_An\\_All-Inclusive\\_Approach\\_to\\_Server-Side\\_Scripting/links/65984fcf0bb2c7472b36211b/Unlocking-PHPs-Potential-An-All-Inclusive-Approach-to-Server-Side-Scripting.pdf](https://www.researchgate.net/profile/ParikshithReddyBaddam/publication/377025333_Unlocking_PHP's_Potential_An_All-Inclusive_Approach_to_Server-Side_Scripting/links/65984fcf0bb2c7472b36211b/Unlocking-PHPs-Potential-An-All-Inclusive-Approach-to-Server-Side-Scripting.pdf)

[57] Course Syllabus and Instructor Plan Web Design II IMED 2315-01 David Trower. (n.d.).

[https://trowertech.com/teaching/IMED\\_2315\\_01.pdf](https://trowertech.com/teaching/IMED_2315_01.pdf)

[58] Afsari, K., Eastman, C.M. and Castro-Lacouture, D. (2017). JavaScript Object Notation (JSON) data serialization for IFC schema in web-based BIM data exchange. Automation in Construction, [online] 77, pp.24-51.

<https://www.sciencedirect.com/science/article/pii/S0926580517300316?via%3Dihub>

[59] hyperskill.org. (n.d.). What is JSON? | Hyperskill University. [online] Available at:

<https://hyperskill.org/university/javascript/what-is-json>

[60] Liu, Z.H., Hammerschmidt, B. and McMahon, D. (2014). JSON data management. Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data.

<https://dl.acm.org/doi/pdf/10.1145/2588555.2595628>

[61] Oracle Help Center. (n.d.). 2 Day + Java Developer's Guide. [online]

<https://docs.oracle.com/en/database/oracle/oracle-database/19/tdpjd/using-java-with-oracle-database.html>

[62] Pratik Patel and Karl Moss, *Java Database Programming with JDBC: Discover the Essentials for Developing Databases for Internet and Intranet Applications*.

<http://biblioteca.isctem.ac.mz/bitstream/123456789/880/1/Java%20Database%20Programming%20with%20JDBC%20Discover%20the%20Essentials%20for%20Developing%20Databases%20for%20Internet%20and%20Intranet%20Applications%20by%20Pratik%20Patel%20Karl%20Moss%20%28z-lib.org%29.pdf>

[63] Coursera. (n.d.). What Is Java Used For? [online] Available at: <https://www.coursera.org/articles/what-is-java-used-for>

[64] Martin Fowler, Patterns of Enterprise Application Architecture. [https://ptgmedia.pearsoncmg.com/images/9780132354790/samplepages/0132354799\\_Sample.pdf](https://ptgmedia.pearsoncmg.com/images/9780132354790/samplepages/0132354799_Sample.pdf)

[65] Jsoup: Java HTML Parser." Baeldung. <https://www.baeldung.com/javawithjsoup#:~:text=Jsoup%20is%20an%20open%2Dsource,to%20parse%20and%20build%20XML.>

[66] soup.org. (n.d.). Use selector-syntax to find elements: jsoup Java HTML parser. [online] . <https://jsoup.org/cookbook/extracting-data/selector-syntax>

[67] Maruyama, H. (2002). XML and Java: Developing Web Applications. [online] Goodle Books. Addison-Wesley Professional. [https://books.google.com.cy/books?hl=en&lr=&id=Qu17E2bsfLUC&oi=fnd&pg=PA1&dq=+javax.xml+is+a+Java+library+that+provides+common+classes+and+interfaces+for+handling+XML+in+Java.++&ots=il\\_4e7p6ed&sig=wOUprLefadc\\_6f09V4sixTaclXA&redir\\_esc=y#v=onepage&q&f=false](https://books.google.com.cy/books?hl=en&lr=&id=Qu17E2bsfLUC&oi=fnd&pg=PA1&dq=+javax.xml+is+a+Java+library+that+provides+common+classes+and+interfaces+for+handling+XML+in+Java.++&ots=il_4e7p6ed&sig=wOUprLefadc_6f09V4sixTaclXA&redir_esc=y#v=onepage&q&f=false)

[68] XML Path Language (XPath) 2.0 (Second Edition). (n.d.) <https://travesia.mcu.es/server/api/core/bitstreams/c2a0f102-671f-4ba0-b550-d4b8ab4e0f83/content>

[69] "Java – Convert String to XML Document and XML Document to String." Baeldung. <https://www.baeldung.com/java-convert-string-xml-dom>

[70] Yi, X., & Lv, X. (2006). "Research and application of E-commerce recommendation system based on collaborative filtering." *Computer Engineering and Applications*, 42(12), 157-159.

<https://www.sciencedirect.com/science/article/abs/pii/S0169023X06000024?via%3Dihb>

[71] Codex, A.C. (2023). Java XML parsing: Reading and manipulating XML documents | Reintech media. [online] reintech.io.

<https://reintech.io/blog/java-xml-parsing-reading-and-manipulating-xml-documents>

[72] Java SAX Parser." Baeldung.

<https://www.baeldung.com/java-sax-parser>

[73] Anon, (n.d.). SAX: Simple API for XML – Itexus. [online] Available at:

<https://itexus.com/glossary/sax-simple-api-for-xml/#gref>

[74] Gaikovina Kula, R., De Roover, C., German, D., Ishio, T. and Inoue, K. (n.d.) Modelling Library Dependencies and Updates in Large Software Repository Universes. [online].

<https://arxiv.org/pdf/1709.04626>

[75] Wu, D. D., Olson, D. L., & Zhao, L. (2013). "Comparison of data mining techniques on customer churn prediction in the telecommunication industry." *Expert Systems with Applications*, 40(2), 463-472.

<https://www.sciencedirect.com/science/article/pii/S0164121223001334>

[76] International Journal Of Multidisciplinary Research Impact Factor: 7.54. (n.d.).

[https://www.ijmrset.com/upload/31\\_An\\_NC.pdf](https://www.ijmrset.com/upload/31_An_NC.pdf)

[77] Ochoa, L., Degueule, T., Falleri, J.-R. and Vinju, J. (2022). Breaking bad? Semantic versioning and impact of breaking changes in Maven Central. *Empirical Software Engineering*, 27(3).

<https://link.springer.com/article/10.1007/s10664-021-10052-y>

[78] Jiang, T., Feng, Y., Wang, J., & Zhang, H. (2022). "An improved firefly algorithm based on chaos search for clustering." *Applied Intelligence*, 52, 2910-2925.  
<https://link.springer.com/content/pdf/10.1007/s10489-022-03864-y.pdf>

# Παράρτημα Α

## A1) Pattern και Matcher

```
Pattern packageNamePattern = Pattern.compile("<name>(.*?)</name>");
Matcher packageNameMatcher = packageNamePattern.matcher(variablesMap.get("responseBody"));

if (packageNameMatcher.find()) {
    variablesMap.put("packageName", packageNameMatcher.group(1));
    // <packagename> tags
    // Further processing with the extracted package number
} else {
    System.out.println("Package number not found");
}

Pattern descriptionPattern = Pattern.compile("<description>(.*?)</description>", Pattern.DOTALL);
Matcher descriptionMatcher = descriptionPattern.matcher(variablesMap.get("responseBody"));
if (descriptionMatcher.find()) {
    variablesMap.put("description", descriptionMatcher.group(1));
    // <description> tags
} else {
    System.out.println("description not found"); //take the description from the pom and after in i
}

// Extracting url
```

Στο Παράρτημα Α θα εξηγήσουμε αναλυτικά λίγο καλύτερα όλη την διαδικασία στην οποία τα δεδομένα διαχειρίζονται από τα pom xml και τοποθετούνται στην βάση δεδομένων. Το pattern χρησιμοποιείται για να εντοπίσει οποιαδήποτε συμφωνία μεταξύ των <name> tags στο responseBody. Αν βρει ένα τέτοιο μοτίβο, η μέθοδος find() του matcher επιστρέφει true, δηλώνοντας ότι βρέθηκε ένα τέτοιο στοιχείο. Αντίστοιχα, αν δε βρει κάποια αντιστοίχιση με το pattern, τότε η find() επιστρέφει false, δηλώνοντας ότι δεν υπάρχει τέτοιο στοιχείο στο responseBody. Σε αυτήν την περίπτωση, εμφανίζεται το μήνυμα "Package number not found". Εάν το find() επιστρέψει true, το πρόγραμμα συνεχίζει με την εκτέλεση της μεθόδου insert Package, περνώντας τα δεδομένα που έχουν εξαχθεί από το responseBody. Αν επιστρέψει false, τότε η μέθοδος insert Package δεν καλείται, αφού δεν βρέθηκε καμία αντιστοίχιση και συνεπώς, δεν υπάρχουν δεδομένα για να εισαχθούν. Ακριβώς το ίδιο ισχύει και για τα <description> tags.

## A2) Μέθοδος Insert στην Java

```
public void insertPackageVersion(int packageId,String version,String URL, String USERNAME, String PASSWORD) {  
    if(!isVersionExists(packageId,version,URL,USERNAME,PASSWORD)) {  
        try (Connection connection = DriverManager.getConnection(URL, USERNAME, PASSWORD)) {  
            String insertPackageQuery = "INSERT INTO packageversiontable (packageId,version) VALUES (?, ?)";  
            PreparedStatement preparedStatement = connection.prepareStatement(insertPackageQuery);  
            preparedStatement.setInt(1, packageId);  
            preparedStatement.setString(2, version);  
            preparedStatement.executeUpdate();  
            System.out.println("Package inserted successfully.");  
            connection.close();  
        } catch (SQLException e) {  
            System.out.println("Insertion failed. Error: " + e.getMessage());  
        }  
    }  
}
```

Αυτή η μέθοδος είναι πολύ σημαντική για το σύστημα γιατί είναι η μέθοδος η οποία παίρνει τα δεδομένα που έρχονται από το pattern και τα βάζει στην βάση δεδομένων. Για αρχή, ελέγχουμε αν υπάρχουν τα στοιχεία είδη στην βάση δεδομένων. Αν δεν υπάρχουν, τότε δημιουργούμε ένα connection με την βάση δεδομένων μας και δημιουργούμε το query το οποίο κάνει το Insert στην βάση. Το PreparedStatement.executeUpdate() χρησιμοποιείται για να εκτελέσει την εντολή SQL και βγάζει το ανάλογο μήνυμα όταν τα δεδομένα μπουκ στην βάση. Μετά από αυτή την διαδικασία κλείνει το connection.



## Παράρτημα Β

Στο προηγούμενο παράρτημα εξηγήσαμε πως τα δεδομένα μεταφέρονται από την Maven Central Repository και μπαίνουν με την μέθοδο Insert στην βάση δεδομένων. Σε αυτό το παράρτημα θα δείξουμε κάποια δεδομένα από την βάση δεδομένων.

| artifact_id     | id             | description                                            | url                                                                                                             |
|-----------------|----------------|--------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|
| java-sdk        | paynow         | This project contains libraries to interface with Z... | <a href="https://github.com/paynow/Paynow-Java-SDK">https://github.com/paynow/Paynow-Java-SDK</a>               |
| user-agent-util | attach-sources | This is a library that can be used to gather infor...  | <a href="https://github.com/youyoudanshui/userAgentUtil">https://github.com/youyoudanshui/userAgentUtil</a>     |
| geocode         | attach-sources | The official Java SDK for the GeoCode Routing ...      | <a href="https://geocode.dev.stefan.zone">https://geocode.dev.stefan.zone</a>                                   |
| web-sheaf       | NULL           | Utility Classes for Building Web Applications for ...  | <a href="https://bitbucket.org/src-zone/web-sheaf.git">https://bitbucket.org/src-zone/web-sheaf.git</a>         |
| sheaf-parent    | checkstyle     | Parent Pom for zone.src.sheaf projects                 | <a href="http://sheaf.src.zone/">http://sheaf.src.zone/</a>                                                     |
| sheaf-deps-bom  | NULL           | Bill of Material for common dependencies of she...     | <a href="https://bitbucket.org/src-zone/sheaf-deps-bom...">https://bitbucket.org/src-zone/sheaf-deps-bom...</a> |
| logback-sheaf   | attach-sources | Logback Classic Module with java-sheaf Initializ...    | <a href="https://bitbucket.org/src-zone/logback-sheaf.git">https://bitbucket.org/src-zone/logback-sheaf.git</a> |
| java-sheaf      | NULL           | Java Utility Classes                                   | <a href="https://bitbucket.org/src-zone/java-sheaf.git">https://bitbucket.org/src-zone/java-sheaf.git</a>       |
| writer          | ossrh          | Writer abstraction for the Refactor Zone Text P...     | <a href="https://refactor.zone/writer">https://refactor.zone/writer</a>                                         |
| test            | ossrh          | Integration tests for the Refactor Zone Text Pr...     | <a href="https://refactor.zone/test">https://refactor.zone/test</a>                                             |
| reader          | ossrh          | Reader abstraction for the Refactor Zone Text ...      | <a href="https://refactor.zone/reader">https://refactor.zone/reader</a>                                         |
| html            | ossrh          | HTML writer for the Refactor Zone Text Processor       | <a href="https://refactor.zone/html">https://refactor.zone/html</a>                                             |

Όπως παρουσιάζεται πιο πάνω, εδώ παρατηρούμε ότι η κάθε γραμμή του πίνακα περιέχει διαφορετικό πακέτο αφού έχουν όλες διαφορετικό “artifact\_id” μεταξύ τους. Σύμφωνα με αυτά τα δεδομένα σημαίνει πως είχαμε 4 pattern αντίστοιχα με τις στήλες του πίνακα (artifact\_id, id, description, url) καθώς και μια insert package table μέθοδο που τα παίρνει όλα αυτά. Στο επόμενο παράρτημα θα εξηγήσουμε πως λειτουργεί ακριβώς η σελίδα και πως παίρνουμε τα δεδομένα από την βάση δεδομένων και τα κάνουμε display στον χρήστη.

## Παράρτημα Γ

Σε αυτό το παράρτημα θα εξηγήσουμε την διαδικασία που παίρνουμε τα δεδομένα από την βάση δεδομένων και τα εμφανίζουμε στην σελίδα.

```
// Create the data object to send in the request body
const data = {
  package_version: package_version,
  group_id: group_id,
  artifact_id: artifact_id,
  lic1: lic1,
  lic2: lic2,
  cName: cName,
  cUrl: cUrl,
  cEmail: cEmail,
  Num1: Num1,
  Num2: Num2,
  ver1: ver1,
  ver2: ver2,
  Dname: Dname,
  Dorganization: Dorganization,
  RepoName: RepoName,
  Repourl: Repourl,
  SrepoName: SrepoName,
  SrepoUrl: SrepoUrl,
  maven_version: maven_version
};
```

Για αρχή, δημιουργούμε ένα αντικείμενο με όλα τα δεδομένα που μπορεί ο χρήστης να τοποθετήσει στη σελίδα.

```
window.location.href = `results.html#` + queryString; //here we send the form data in results.html
```

Μετά από αυτό στέλνουμε τα δεδομένα αυτά στην results.html όπου είναι η σελίδα στην οποία θα εμφανιστούν τα δεδομένα.

```
<div id="dataContainer"></div> <!-- This is where the fetched data will be displayed -->
```

Παρουσιάζεται το div στην results.html όπου θα μουν τα δεδομένα που θα έρθουν από την βάση.

```
// Function to update content based on current page
function updateContent() {
  // Retrieve query parameters from the URL
  const urlParams = new URLSearchParams(window.location.hash.slice(1)); // Αφαίρεση του "#" από την αρχή

  // Extract data from query parameters
  const data = {}; //here we have all the data from maven.html
  urlParams.forEach((value, key) => {
    data[key] = value;
  });

  // Call function to fetch data from server
  fetchData(data);
}
```

Ενώνουμε την Maven.js με την results.js και βάζουμε το αντικείμενο data με όλα τα δεδομένα που μπορεί ο χρήστης να βάλει στα πεδία. Τέλος καλούμε την fetch Data με τα δεδομένα η οποία θα στείλει ένα αίτημα GET στον χρήστη και αν το response είναι 'ok' θα πάρει τα δεδομένα αλλιώς θα βγάλει μήνυμα λάθους.

```
// Function to fetch data from server
function fetchData(data) {
  // Send a GET request to the server with the query parameters
  console.log('Form data:', data); // Log the fetched data

  const url = `server.php?${new URLSearchParams(data).toString()}`; //here we send tha data to php
```

Η μέθοδος fetchData όπου τα δεδομένα στέλνονται σε Strings στην php για να τα επεξεργαστεί και να φέρει πίσω τα δεδομένα που θα ψάξει ο χρήστης.

```

2 // Establish database connection
3 $servername = "localhost";
4 $username = "root";
5 $password = "Maven12CY#";
6 $database = "mavendatabase";
7
8 // Create connection
9 $conn = new mysqli($servername, $username, $password, $database);
10
11 // Check connection
12 if ($conn->connect_error) {
13     die("Connection failed: " . $conn->connect_error);
14 }
15
16 // Handle POST request to search for data
17 if ($_SERVER['REQUEST_METHOD'] === 'GET') { //here we get the data from results.js and use this to make the select
18     // Extract data from the request
19     $package_version = $_GET['package_version'] ?? '';
20     $group_id = $_GET['group_id'] ?? '';
21     $artifact_id = $_GET['artifact_id'] ?? '';
22     $lic1 = $_GET['lic1'] ?? '';
23     $lic2 = $_GET['lic2'] ?? '';
24     $cName = $_GET['cName'] ?? '';
25     $cUrl = $_GET['cUrl'] ?? '';
26     $cEmail = $_GET['cEmail'] ?? '';
27     $Num1 = $_GET['Num1'] ?? '';
28     $Num2 = $_GET['Num2'] ?? '';
29     $ver1 = $_GET['ver1'] ?? '';
30     $ver2 = $_GET['ver2'] ?? '';
31     $Dname = $_GET['Dname'] ?? '';
32     $Dorganization = $_GET['Dorganization'] ?? '';
33     $RepoName = $_GET['RepoName'] ?? '';
34     $Repourl = $_GET['Repourl'] ?? '';
35     $SrepoName = $_GET['SrepoName'] ?? '';
36     $SrepoUrl = $_GET['SrepoUrl'] ?? '';
37     $maven_version = $_GET['maven_version'] ?? '';
38 }

```

Αφού στείλουμε τα δεδομένα στην php, ενώνουμε την php με την βάση δεδομένων και με το `if ($_SERVER['REQUEST_METHOD']=='GET')` παίρνουμε τα δεδομένα από την results.js.

```

// Construct the SQL query dynamically based on provided parameters
$sql = "SELECT DISTINCT packagetable.packageId
FROM packagetable
LEFT JOIN licensetable ON packagetable.packageId = licensetable.packageId
LEFT JOIN contributortable ON packagetable.packageId = contributortable.packageId
LEFT JOIN developertable ON packagetable.packageId = developertable.packageId
LEFT JOIN distributionmanagmenttable ON packagetable.packageId = distributionmanagmenttable.packageId
LEFT JOIN repositorytable ON distributionmanagmenttable.distributionmanagmentid = repositorytable.dist
LEFT JOIN snapshotrepositorytable ON distributionmanagmenttable.distributionmanagmentid = snapshotrepc
LEFT JOIN prerequisitestable ON packagetable.packageId = prerequisitestable.packageId
LEFT JOIN dependencytable ON packagetable.packageId = dependencytable.packageId
WHERE 1 = 1"; // Start with a true condition

// Add other conditions based on provided parameters
if ($package_version) $sql .= " AND packagetable.package_version = '$package_version'";
if ($group_id) $sql .= " AND packagetable.group_id LIKE '%$group_id%'";
if ($artifact_id) $sql .= " AND packagetable.artifact_id LIKE '%$artifact_id%'";
if ($lic1) $sql .= " AND licensetable.name LIKE '%$lic1%'";
if ($lic2) $sql .= " AND licensetable.url LIKE '%$lic2%'";
if ($cName) $sql .= " AND contributortable.name LIKE '%$cName%'";
if ($cUrl) $sql .= " AND contributortable.url LIKE '%$cUrl%'";
if ($cEmail) $sql .= " AND contributortable.email LIKE '%$cEmail%'";
if ($Dname) $sql .= " AND developertable.name LIKE '%$Dname%'";
if ($Dorganization) $sql .= " AND developertable.organization LIKE '%$Dorganization%'";
if ($RepoName) $sql .= " AND repositorytable.name LIKE '%$RepoName%'";
if ($Repourl) $sql .= " AND repositorytable.url LIKE '%$Repourl%'";
if ($SrepoName) $sql .= " AND snapshotrepositorytable.name LIKE '%$SrepoName%'";
if ($SrepoUrl) $sql .= " AND snapshotrepositorytable.url LIKE '%$SrepoUrl%'";
if ($maven_version) $sql .= " AND prerequisitestable.maven_version LIKE '%$maven_version%'";

```

Το \$sql είναι ένα ερώτημα SQL που χρησιμοποιείται για να επιστρέψει μια λίστα μοναδικών packageId που πληρούν ορισμένες συνθήκες. Αυτές οι συνθήκες περιέχονται στο \$sql ως σειρά WHERE δηλώσεων, με την προσθήκη ορισμένων συνθηκών βάσει των παρεχόμενων παραμέτρων, όπως η έκδοση του πακέτου, το groupId, το artifactId, και άλλα. Σε αυτό το σημείο κάνουμε SELECT τα package\_id όλων των πινάκων για να μπορούμε στο επόμενο βήμα να επεξεργαστούμε τις πληροφορίες που περιέχονται στο κάθε πίνακα. Στα if πιο κάτω ελέγχουμε αν στο συγκεκριμένο πεδίο τοποθέτησε κάτι ο χρήστης. Αν τοποθέτησε θα καλεστεί η sql για το συγκεκριμένο πεδίο. Το LIKE σημαίνει ότι ο χρήστης δεν αναγκάζεται να τοποθετήσει ολόκληρη την λέξη στο πεδίο αλλά κάτι που την περιέχει για παράδειγμα( abb θα του βγάλει abbot). Επίσης με το AND τονίζουμε ότι ο χρήστης μπορεί να τοποθετήσει πολλές τιμές σε πολλά πεδία και όταν γίνει η αναζήτηση θα καλεστούν όλες μαζί.

```

// Query to get detailed information based on packageIds
$detailedSql = "SELECT packagetable.packageId,packagetable.package_name, packagetable.package_version, packageta
licensetable.name AS LicenseName, licensetable.url AS LicenseUrl,
contributortable.name AS contributorName, developertable.name AS DeveloperName, developertable.o
repositorytable.name AS RepositoryName, repositorytable.url AS RepositoryUrl,
(SELECT COUNT(*) FROM contributortable WHERE contributortable.packageId = packagetable.packageId)
(SELECT COUNT(*) FROM packageversiontable WHERE packageversiontable.packageId = packagetable.pac
FROM packagetable
LEFT JOIN licensetable ON packagetable.packageId = licensetable.packageId
LEFT JOIN contributortable ON packagetable.packageId = contributortable.packageId
LEFT JOIN packageversiontable ON packagetable.packageId = packageversiontable.packageId
LEFT JOIN developertable ON packagetable.packageId = developertable.packageId
LEFT JOIN distributionmanagmenttable ON packagetable.packageId = distributionmanagmenttable.packa
LEFT JOIN repositorytable ON distributionmanagmenttable.distributionmanagmentid = repositorytable
WHERE packagetable.packageId IN (" . implode(',', $packageIds) . ")";

// Add checks for Num1 and Num2
if (is_numeric($Num1) && is_numeric($Num2)) {
    $detailedSql .= " AND (SELECT COUNT(*) FROM contributortable WHERE contributortable.packageId = packagetable
} elseif (is_numeric($Num1)) {
    $detailedSql .= " AND (SELECT COUNT(*) FROM contributortable WHERE contributortable.packageId = packagetable
} elseif (is_numeric($Num2)) {
    $detailedSql .= " AND (SELECT COUNT(*) FROM contributortable WHERE contributortable.packageId = packagetable
}

if (is_numeric($ver1) && is_numeric($ver2)) {
    $detailedSql .= " AND (SELECT COUNT(*) FROM packageversiontable WHERE packageversiontable.packageId = packag
} elseif (is_numeric($ver1)) {
    $detailedSql .= " AND (SELECT COUNT(*) FROM packageversiontable WHERE packageversiontable.packageId = packag
} elseif (is_numeric($ver2)) {
    $detailedSql .= " AND (SELECT COUNT(*) FROM packageversiontable WHERE packageversiontable.packageId = packag
}

```

Στο τμήμα \$detailedSql, παίρνουμε τα μοναδικά packageId που πληρούν τις προϋποθέσεις από το πρώτο ερώτημα και στη συνέχεια χρησιμοποιούμε αυτά τα packageId για να ανακτήσουμε λεπτομερείς πληροφορίες για κάθε πακέτο, όπως το package\_name, η package\_version, το group\_id, το artifact\_id, και άλλα, από πολλούς πίνακες μέσω διαφόρων συνδέσεων (LEFT JOIN). Η SQL ερώτηση συνδέει τους πίνακες μεταξύ τους χρησιμοποιώντας το πεδίο packageId ως κλειδί σύνδεσης. Όταν εκτελεστεί η detailedSql, περνάει τη λίστα των μοναδικών packageId που προέκυψαν από το προηγούμενο ερώτημα στην IN συνθήκη. Έτσι, η detailedSql επιστρέφει όλες τις λεπτομέρειες για τα packageId που πληρούν τις προϋποθέσεις που έχουν οριστεί στο \$sql. Οι πίνακες είναι συνδεδεμένοι μεταξύ τους μέσω των LEFT JOIN, επιτρέποντας την ανάκτηση των αντίστοιχων εγγραφών ακόμη και εάν δεν υπάρχουν αντίστοιχες εγγραφές σε όλους τους συνδεδεμένους πίνακες. Αυτό εξασφαλίζει ότι κανένα δεδομένο δεν χάνεται και παίρνουμε πλήρες σύνολο αποτελεσμάτων από το ερώτημα.



```
// Return the JSON response
header('Content-Type: application/json');
echo json_encode($response); //here we take the packets from mysql an put them in results js for display
}
```

Αφού πάρουμε τα δεδομένα από την βάση δεδομένων τα στέλνουμε στην results.js για να τα εμφανίσει στον χρήστη. Στην περίπτωση του GET από την βάση κάνουμε πάντα json\_encode.

```
fetch(url)
  .then(response => {
    if (!response.ok) {
      throw new Error('Network response was not ok');
    }
    return response.json(); // Parse response as JSON
  })
  .then(jsonData => { //here we have all tha packets from php
    // Store fetched data globally
    console.log('Fetched data:', jsonData); // Log the fetched data

    fetchedData = jsonData; // Parse response as JSON
    // Call function to display data on the page
    displayData(); //here we do display the data in results.html
  })
  .catch(error => console.error('Error fetching data:', error));
}
```

Τέλος, εμφανίζουμε τα δεδομένα στην results.html αφού τα μετατρέψουμε πρώτα σε json και παράλληλα ο χρήστης έχει την πρόσβαση να αναγνωρίσει τα πακέτα ανάλογα με τα πεδία που έχει τοποθετήσει.