

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ  
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«SERVER DATA BACKUP MANAGEMENT»

****

| **Του φοιτητή**  **ΕΜΜΑΝΟΥΗΛΙΔΗ ΣΟΦΟΚΛΗ**  **Αρ. Μητρώου: 164659** | **Επιβλέπων**  **Ονοματεπώνυμο ΟΥΓΙΑΡΟΓΛΟΥ ΣΤΕΦΑΝΟΣ**  **Βαθμίδα Καθηγητής** |
| --- | --- |

**Ημερομηνία ……………**

Server Data Backup Management

Κωδικός Δ.Ε. …

Εμμανουηλίδης Σοφοκλής

Ονοματεπώνυμο εισηγητή …

Ημερομηνία ανάληψης Δ.Ε. …

Ημερομηνία περάτωσης Δ.Ε. …

*Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.*

*Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Εμμανουηλίδη Σοφοκλή που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ’ οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.*

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραιτήτως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

# Αφιέρωση

Την πτυχιακή μου εργασία την αφιερώνω σε όλη μου την οικογένεια που όλα αυτά τα χρόνια είναι στο πλάι μου και με στηρίζει σε κάθε στιγμή της ζωής μου και εύχομαι η εργασία αυτή στο μέλλον να βοηθήσει όποιον την χρειαστεί στις εργασίες και απαιτήσεις τους και να τους διευκολύνει στην καθημερινότητα τους.

# Πρόλογος

Ο κυριότερος λόγος που επέλεξα αυτό το θέμα της πτυχιακής είναι η αυξημένη ανάγκη στις μέρες μας για την ύπαρξη αντιγράφων ασφαλείας βάσεων δεδομένων. Στην σημερινή εποχή όπου όλα πλέον αρχίζουν να γίνονται ψηφιοποιημένα και να μειώνεται η χρήση του χάρτινου αρχείου, η απώλεια δεδομένων μπορεί να έχει σοβαρές επιπτώσεις και ατομικά στους χρήστες και ιδιαίτερα στις επιχειρήσεις. Η διαχείριση ενός αξιόπιστου συστήματος backup είναι καθημερινή και σημαντική υπόθεση για την αποφυγή αστοχιών συστήματος, κυβερνοεπιθέσεων ή ανθρώπινων λαθών.

Αυτόν τον στόχο επιτυγχάνει η δημιουργία μιας πλατφόρμας όπου οι χρήστες μπορούν να πραγματοποιούν πολλαπλά και ταυτόχρονα backup βάσεων δεδομένων από διαφορετικά domains και να αποθηκεύονται τόσο τα αρχεία που δημιουργούνται, όσο και τα στοιχεία αυθεντικοποίησης τους αποθηκευμένα με αποκρυπτογράφηση στην βάση δεδομένων budbserver του παρόχου nirea.iee.ihu.gr.

Αυτή η πλατφόρμα δίνει την δυνατότητα στους χρήστες να πραγματοποιούν πολλαπλά backups απο διαφορετικούς παρόχους και βάσεις δεδομένων με ένα κουμπί , γλιτώνοντας έτσι πολύτιμο χρόνο και πολλαπλά logins και ssh.

# Περίληψη

Το θέμα - ζητούμενο της παρούσας πτυχιακής εργασίας είναι να μπορούν οι χρήστες μέσω μιας ιστοσελίδας να πραγματοποιούν πολλαπλά backups σε πολυάριθμες βάσεις δεδομένων από τους ίδιους ή και διαφορετικούς παρόχους, γλιτώνοντας έτσι την ξεχωριστή αυθεντικοποίηση για κάθε μια βάση δεδομένων και πάροχο αντίστοιχα και κερδίζοντας πολύτιμο χρόνο. Όλα αυτά τα στοιχεία σύνδεσης και αυθεντικοποίησης αποθηκεύονται σε ξεχωριστό για κάθε χρήστη προφίλ με μοναδικό email ως username για τον καθένα αντίστοιχα.

Τα κυριότερα εργαλεία που χρησιμοποιήθηκαν για την πραγματοποίηση των αντιγράφων ασφαλείας είναι:

1. MYSQLDUMP για την δημιουργία αντιγράφων ασφαλείας στις βάσεις δεδομένων mysql.
2. PG\_DUMP για την δημιουργία αντιγράφων ασφαλείας στις βάσεις δεδομένων postgres..

Το δύσκολο κομμάτι στην υλοποίηση αυτής της πλατφόρμας ήταν οι διαφορές μεταξύ των απαιτήσεων του κάθε παρόχου στους τομείς της αυθεντικοποίησης και της δημιουργίας αρχείου backup. Προκειμένου να αντιμετωπίσω αυτή την ποικιλία διαφορετικών απαιτήσεων, δημιούργησα ένα σύστημα το οποίο υποστηρίζει ssh tunneling για ασφαλή επικοινωνία, καθώς και την υποστήριξη διαφορετικών ειδών βάσεων δεδομένων (mysql , postgres).

Προκειμένου να πετύχω την αλληλεπίδραση της εφαρμογής με την βάση δεδομένων χρησιμοποίησα PHP (PHP: Hypertext Preprocessor). Για την δομή και την οργάνωση του περιεχομένου της ιστοσελίδας αντίστοιχα χρησιμοποίησα HTML (Hyper Text Markup Language) και για την εμφάνιση και τον σχεδιασμό του περιβάλλοντος χρήστη χρησιμοποίησα CSS (Cascading Style Sheets).

Με την χρήση και τον συνδυασμό όλων των παραπάνω τεχνολογιών, δημιούργησα ένα λειτουργικό και δυναμικό σύστημα δημιουργίας αντιγράφων ασφαλείας βάσεων δεδομένων.

# Abstract

The purpose of this THESIS project is to allow users to perform multiple backups of numerous databases from the same OR different providers through a website, both for saving the need for separate authentication for each database and provider respectively and saving valuable time. All of these login and authentication details are stored in a separate unique profile for each user with a unique email as username for each one of these profiles respectively.

The main tools that are used to perform the backups are:

* MYSQLDUMP for backing up the ‘mysql’ databases
* PG\_DUMP for backing up the ‘postgres’ databases

The most challenging parts in creating this platform and making it operational were the numerous differences between the requirements of each provider in the areas of authentication and backup file creation. In order to address this variety of different requirements, I created a system that supports ssh tunneling for secure communication and also supports different kinds of databases (mysql, postgres).

In order to achieve the interaction between the application and the database I used PHP (PHP: Hypertext Preprocessor). For the structure and the organization of the website content respectively I used HTML (Hyper Text Markup Language). For the appearance and the design of the UI I used CSS (Cascading Style Sheets).

By using and combining all of the above technologies, I created a functional, dynamic and user friendly database backup system.

«BACKUP DATABASE SERVER (BUDS)»

«SOFOKLIS EMMANOUILIDIS»

# Ευχαριστίες

Ένα τεράστιο ευχαριστώ οφείλω στον κ. Στέφανο Ουγιάρογλου, ο οποίος μέσα στην καρδιά του καλοκαιριού επικοινώνησε μαζί μου, είδε το ενδιαφέρον μου για πτυχιακή εργασία, με ανέλαβε και σε όλη την διάρκεια της είναι παρών με συμβουλές και ιδέες για βελτιώσεις. Επίσης θα ήθελα να ευχαριστήσω και τον κ. Σιδηρόπουλο Αντώνιο για την επίλυση αποριών και εύρεση λύσεων σε κομμάτια του κώδικα του προγράμματος μεγάλης σημασίας για την λειτουργικότητα του.

# Περιεχόμενα

[Αφιέρωση 4](#_heading=h.fxps2crewnav)

[Πρόλογος 5](#_heading=h.v0d840xnf9xv)

[Περίληψη 6](#_heading=h.nzjn9x2p7k7s)

[Abstract 7](#_heading=h.edpobdh8jbge)

[Ευχαριστίες 9](#_heading=h.xrjxgrpogb92)

[Περιεχόμενα 10](#_heading=h.h390x92gcwp1)

[Συντομογραφίες 12](#_heading=h.sudojwd0l3qw)

[Διαγράμματα Ροής - Flowcharts 12](#_heading=h.80xvuehxm925)

[Κεφάλαιο 1ο: Ασφάλεια Δεδομένων 1](#_heading=h.m4ginymeyamo)

[1.1 Εισαγωγή 1](#_heading=h.jhwwiwt3g28j)

[1.2 Αντίγραφα Ασφαλείας Βάσεων Δεδομένων 1](#_heading=h.rlbcty8dkamp)

[1.3 Κίνητρο 3](#_heading=h.nc1jdbompdjh)

[1.4 Συνεισφορά 3](#_heading=h.hcj4t5hdd68t)

[1.5 Οργάνωση εργασίας 4](#_heading=h.hlskkw5wfyd8)

[Κεφάλαιο 2ο: Τύποι Αντιγράφων Ασφαλείας 5](#_heading=h.epn31d2ek40h)

[2.1 Εισαγωγή 5](#_heading=h.vh5u3ga5anz9)

[2.2 FULL BACKUP - Πλήρες Αντίγραφο Ασφαλείας 5](#_heading=h.2xrk3lmwzw0m)

[2.2.1 Δημιουργία full backup με την χρήση MySQL 5](#_heading=h.qgq3veiqachv)

[2.2.2 Δημιουργία full backup με την χρήση PostgreSQL 6](#_heading=h.3tqk4x8n71zt)

[2.3 INCREMENTAL BACKUP - Επαυξητικό Αντίγραφο Ασφαλείας 7](#_heading=h.v5700o47v19x)

[2.3.1 Δημιουργία incremental backup με την χρήση MySQL 8](#_heading=h.onyiytnl2mal)

[2.3.2 Δημιουργία incremental backup με την χρήση PostgreSQL 8](#_heading=h.eusj6cfdzatf)

[2.4 DIFFERENTIAL BACKUP - Διαφορικό Αντίγραφο Ασφαλείας 10](#_heading=h.6xc4wupfxhs0)

[2.4.1 Δημιουργία differential backup με την χρήση MySQL 10](#_heading=h.gslf72oetd44)

[2.4.2 Δημιουργία differential backup με την χρήση PostgreSQL 11](#_heading=h.o93x4ht0k1wa)

[2.5 SNAPSHOT BACKUP - Αντίγραφο Ασφαλείας Στιγμιοτύπου 12](#_heading=h.exwpuvrs75a0)

[2.5.1 Δημιουργία snapshot backup με την χρήση MySQL 12](#_heading=h.bpyqhq177q5k)

[2.5.2 Δημιουργία snapshot backup με την χρήση PostgreSQL 13](#_heading=h.z9olgx8694ir)

[Κεφάλαιο 3ο: Τεχνολογίες 14](#_heading=h.bsqvorattqtp)

[3.1 HTML 14](#_heading=h.58noipd4x2s0)

[3.1.1 Θετικά χαρακτηριστικά της HTML 14](#_heading=h.4db0cslo61st)

[3.1.2 Αρνητικά χαρακτηριστικά της HTML 14](#_heading=h.dmvv1y5igoqg)

[3.1.3 Συμπέρασμα και αιτιολόγηση επιλογής της HTML 15](#_heading=h.m0n1t6c0lkvb)

[3.2 CSS 16](#_heading=h.783tkz7cv6mb)

[3.2.1 Θετικά χαρακτηριστικά της CSS 16](#_heading=h.4bkxkpez4oya)

[3.2.2 Αρνητικά χαρακτηριστικά της CSS 16](#_heading=h.fbu0gs67tqte)

[3.3 PHP 17](#_heading=h.fjul7igud9gh)

[3.3.1 Θετικά χαρακτηριστικά της PHP 17](#_heading=h.95riaem4befg)

[3.3.2 Αρνητικά χαρακτηριστικά της PHP 18](#_heading=h.68jnm5ur4ctk)

[3.4 SQL 18](#_heading=h.68a57reiaclm)

[3.4.1 Θετικά χαρακτηριστικά της SQL 19](#_heading=h.qbmc1ov440ql)

[Κεφάλαιο 4ο: SQL 19](#_heading=h.hq3fqt9v0bk4)

[4.1 Μαθηματικές σχέσεις 19](#_heading=h.z8yytyuo9y95)

[4.2 Βιβλιογραφικές αναφορές 19](#_heading=h.nobz5mcl7dr6)

[4.3 Επίλογος 20](#_heading=h.ceysfzqgcgi4)

[Κεφάλαιο 5ο: Συμπεράσματα ή/και προτάσεις βελτίωσης 21](#_heading=h.xxt1nwa9ef16)

[ΒΙΒΛΙΟΓΡΑΦΙΑ 23](#_heading=h.3d3p52rgmsac)

# Συντομογραφίες

Δ.Ε. Διπλωματική Εργασία

ΔΙΠΑΕ Διεθνές Πανεπιστήμιο Ελλάδος

Π.Ε. Πτυχιακή Εργασία

# **Δι**α**γρ**ά**μματα Ροής - Flowchart**s

# Ασφάλεια Δεδομένων

## Εισαγωγή

[1] [3] Στον σύγχρονο ψηφιακό κόσμο, σταδιακά αρχίζει να μειώνεται η χρήση του χαρτιού και γενικότερα του υλικού μέσου αποθήκευσης οποιουδήποτε είδους πληροφορίας, δεδομένου, αρχείου κτλ. Τα δεδομένα πλέον αποθηκεύονται ψηφιακά σε βάσεις δεδομένων, παρέχοντας έτσι την δυνατότητα στον κόσμο να έχει ευκολότερη και γρηγορότερη πρόσβαση στις πληροφορίες. Ταυτόχρονα αυξάνεται σε μεγάλο βαθμό η εξοικονόμηση του χώρου.

[2] Παρότι όμως υπάρχουν αρκετά οφέλη με την ψηφιακή αποθήκευση δεδομένων, η ανάγκη για την ύπαρξη αντιγράφων ασφαλείας παραμένει σημαντική και αμετάβλητη. Όπως παλαιότερα διατηρούσαμε φυσικά αντίγραφα για προστασία από οποιαδήποτε μορφή απώλειας, έτσι και σήμερα η δημιουργία αντιγράφων ασφαλείας (backup) είναι απαραίτητη για την ασφάλεια των ψηφιακών δεδομένων.

Η απώλεια ή ακόμα και η καταστροφή ψηφιακών αρχείων, η δυσλειτουργία και η μη προσβασιμότητα σε βάσεις δεδομένων μπορεί να έχει σοβαρές συνέπειες είτε για τον χρήστη ατομικά είτε και για ολόκληρες επιχειρήσεις, καθιστώντας έτσι την ύπαρξη αντιγράφων ασφαλείας κρίσιμα για την επιχειρησιακή συνέχεια και την προστασία των πληροφοριών και δεδομένων.

## Αντίγραφα Ασφαλείας Βάσεων Δεδομένων

Τα 2 βασικά εργαλεία που χρησιμοποιήθηκαν για την δημιουργία αντιγράφων ασφαλείας στις βάσεις δεδομένων **MySQL** και **PostgreSQL**, είναι το **mysqldump** και το **pg\_dump** αντίστοιχα. Καθένα από αυτά τα εργαλεία έχει τα δικά του πλεονεκτήματα και μειονεκτήματα, ο συνδυασμός των οποίων επηρεάζουν την επιλογή τους από τον κάθε χρήστη, καθοριζόμενη από τις εκάστοτε ανάγκες του.

* **mysqldump**: Είναι σημαντικό μέρος του συνόλου εντολών της MySQL και χρησιμοποιείται για την δημιουργία λογικών αντιγράφων ασφαλείας. Μέσω του mysqldump, οι χρήστες μπορούν να εξάγουν δομές και δεδομένα βάσεων δεδομένων σε μορφή SQL, έχοντας έτσι την δυνατότητα αναπαραγωγής τους σε έναν άλλο διακομιστή ή απλά για σκοπούς αποθήκευσης. [4]

* **pg\_dump**: Είναι το αντίστοιχο εργαλείο για την PostgreSQL, επιτρέποντας την δημιουργία αντιγράφων ασφαλείας μιας βάσης δεδομένων σε μορφή SQL script. Το pg\_dump εξασφαλίζει ότι τα αντίγραφα θα είναι πλήρη και ενημερωμένα, ακόμα και όταν η βάση δεδομένων χρησιμοποιείται ενεργά. [5]

ΠΛΕΟΝΕΚΤΗΜΑΤΑ

* mysqldump:
  + Ευκολία χρήσης και ευρεία υποστήριξης.
  + Δυνατότητα εξαγωγής δεδομένων αντιγράφου ασφαλείας σε διάφορες μορφές, όπως SQL, CSV και XML.
  + Κατάλληλο για χρήση σε μικρές εώς μεσαίες βάσεις δεδομένων.[7]
* pg\_dump:
  + Δημιουργία συνεπών αντιγράφων, ακόμη και κατά την διάρκεια ενεργής χρήσης της βάσης δεδομένων.
  + Υποστήριξη διαφόρων μορφών εξαγωγής, όπως plain-text SQL script και archive files.[5][6]
  + Επιτρέπει την επιλεκτική ανάκτηση δεδομένων μέσω της εντολής pg\_restore.[6]

MEIONEKTHMATA

* mysqldump:
  + Για πολύ μεγάλες βάσεις δεδομένων μπορεί να είναι αργό λόγω της μονοπυρηνικής επεξεργασίας δεδομένων.
  + Αντίστοιχα η διαδικασία ανάκτησης δεδομένων μπορεί να είναι ιδιαίτερα χρονοβόρα για πολύ μεγάλα σύνολα δεδομένων.[7]
* pg\_dump:
  + Μπορεί να δημιουργήσει αντίγραφο ασφαλείας για μόνο μια βάση δεδομένων και όχι για πολλές ταυτόχρονα.[9]
  + Η δημιουργία αντιγράφων ασφαλείας ενός συστήματος αρχείων λειτουργεί μόνο για την πλήρη δημιουργία αντιγράφων ασφαλείας και επαναφορά ολόκληρης συστάδας βάσεων δεδομένων. Αυτό σημαίνει ότι η δημιουργία αντιγράφων ασφαλείας και η επαναφορά ορισμένων μεμονωμένων αρχείων ή πινάκων αντίστοιχα είναι αδύνατη.[8]

Λαμβάνοντας υπόψη όλα τα παραπάνω δεδομένα. καταλήγουμε στο συμπέρασμα ότι η επιλογή μεταξύ mysqldump και pg\_dump εξαρτάται από τις ανάγκες του κάθε συστήματος και τα χαρακτηριστικά της κάθε βάσης δεδομένων. Η Mysql είναι προτιμότερο να χρησιμοποιείται για μικρότερες βάσεις δεδομένων και είναι πιο απλή και γρήγορη. Ωστόσο όταν πρόκειται για μεγαλύτερες βάσεις δεδομένων και για πιο πολύπλοκα δεδομένα, εκεί προτιμάται η χρήση της PostgreSQL ,η οποία προσφέρει συνεπή backups ακόμα και σε ενεργά συστήματα. Η διαχείριση της ωστόσο απαιτεί μεγαλύτερη κατανόηση και είναι πιο δύσκολη στη χρήση. Συνεπώς, η επιλογή εξαρτάται από το μέγεθος, την πολυπλοκότητα αλλά και τις ανάγκες ανάκτησης των δεδομένων.

## Κίνητρο

Το κίνητρο μου για να δημιουργήσω αυτή την web εφαρμογή ήταν να διευκολύνω την δημιουργία αντιγράφων ασφαλείας από πολυάριθμες βάσεις δεδομένων από διαφορετικούς παρόχους , χωρίς να απαιτείται από τον χρήστη να συνδέεται ξεχωριστά με πολύπλοκες διαδικασίες σε κάθε μια από αυτές. Το μέγεθος του κινδύνου που μπορεί να επιφέρει στους χρήστες και στις επιχειρήσεις αντίστοιχα η απώλεια δεδομένων σε συνδυασμό με την ανάγκη για την ύπαρξη ενός πιο γρήγορου τρόπου δημιουργίας αντιγράφων, ήταν 2 σύγχρονα προβλήματα που ήθελα να τα λύσω με αυτή την εφαρμογή.

## Συνεισφορά

Η τελική συνεισφορά της δημιουργίας μια web εφαρμογής, η οποία δημιουργεί backups βάσεων δεδομένων σε ξεχωριστά προφίλ είναι η εξασφάλιση της ταχύτητας, της ευκολίας και ειδικότερα της ασφάλειας των δεδομένων. Αυτή η εφαρμογή δίνει την δυνατότητα σε κάθε χρήστη να έχει απομακρυσμένη πρόσβαση στα δικά του αντίγραφα ασφαλείας,, χωρίς να χρειαστεί να ανησυχεί για απώλειες ή για περίπλοκες διαδικασίες ανάκτησης. Αυτό συμβάλλει και στην μείωση του χρόνου αποκατάστασης των δεδομένων, καθώς και στην καλύτερη οργάνωση και εμπειρία του χρήστη. Επιπλέον η αυτοματοποίηση της διαδικασίας της αυθεντικοποίησης για άμεση πρόσβαση στις βάσεις δεδομένων του χρήστη μειώνει σημαντικά τον χρόνο που απαιτείται για όλη την διαδικασία αυτή και ταυτόχρονα εξασφαλίζει ότι τα δεδομένα θα είναι πάντα διαθέσιμα, βελτιώνοντας έτσι την αξιοπιστία του συστήματος και μειώνοντας το ανθρώπινο λάθος σε μεγάλο βαθμό.

## Οργάνωση εργασίας

Η εργασία αυτή οργανώνεται από τα στάδια που θα αναλυθούν παρακάτω, όπου με μια αλληλουχία βημάτων αναπτύχθηκε μια εφαρμογή δημιουργίας αντιγράφων ασφαλείας βάσεων δεδομένων.

Αρχικά, υλοποιήθηκαν οι βασικές λειτουργίες αυθεντικοποίησης χρηστών, με τον συνδυασμών των λειτουργιών σύνδεσης υπάρχοντος χρήστη (Login), εγγραφής καινούργιου χρήστη (Sign Up) και επαναφοράς κωδικού πρόσβασης (Forgot Password Reset) σε μία φόρμα. Έτσι εξασφαλίστηκε η ασφαλή και η άμεση πρόσβαση στην εφαρμογή και στα δεδομένα του κάθε χρήστη ξεχωριστά.

Στην συνέχεια αναπτύχθηκε η φόρμα δημιουργίας νέων συνδέσεων (New Database Connection), όπου ο χρήστης μπορεί να αποθηκεύει στοιχεία αυθεντικοποίησης για διαφορετικές τοπικές και απομακρυσμένες (μέσω ssh tunneling) βάσεις δεδομένων. Ταυτόχρονα ο χρήστης μπορεί κατά την δημιουργία της καινούργιας σύνδεσης και το είδος της βάσης που επιθυμεί να αποθηκεύσει (PostgreSQL , MySQL) , για να χρησιμοποιηθούν οι αντίστοιχες εντολές δημιουργίας αντιγράφων ασφαλείας pg\_dump και mysqldump.

Προτού ωστόσο αποθηκευτούν όλες αυτές οι διαφορετικές συνδέσεις, πραγματοποιούνται δοκιμές εγκυρότητας για να διασφαλιστεί η σωστή λειτουργία των συνδέσεων και να αποφευχθούν τα λάθη στις απόπειρες δημιουργίας αντιγράφων ασφαλείας.

Η αποθήκευση των αντιγράφων ασφαλείας που δημιουργεί ο χρήστης με το που καταφέρει να αποθηκεύσει επιτυχώς τις συνδέσεις του γίνεται στην βάση δεδομένων της ιστοσελίδας στο αντίστοιχο του προφίλ με την δυνατότητα και για τοπική αποθήκευση, διασφαλίζοντας τον χρήστη ότι μπορεί να τα κατεβάσει όποτε επιθυμεί.

Τέλος, όλο αυτό το περιβάλλον περιήγησης χρήστη εμπλουτίστηκε με την χρήση CSS, κάνοντας το πιο ελκυστικό και ευχάριστο, με ευδιάκριτα κουμπιά και ομαλή διάταξη των δεδομένων στην οθόνη του χρήστη. Παράλληλα, συνδυάζοντας HTML και PHP και με την χρήση των εντολών SQL για την επικοινωνία με τους servers, δημιουργήθηκε μια δομημένη και λειτουργική εφαρμογή, προσφέροντας ομαλή εμπειρία χρήστη και εξασφαλίζοντας του την δυνατότητα να κάνει εύκολα και γρήγορα backups και να τα διαχειρίζεται και να τα αποθηκεύει όπως και όπου αυτός επιθυμεί.

# Τύποι Αντιγράφων Ασφαλείας

## Εισαγωγή

Ανάλογα με τις ανάγκες του κάθε συστήματος και των χρηστών την κάθε χρονική στιγμή, υπάρχουν 4 διαφορετικά είδη αντιγράφων ασφαλείας. Αυτά τα 4 είδη backup (Full, Incremental, Differential και Snapshot) επιφέρουν διαφορετικά πλεονεκτήματα και μειονεκτήματα αντίστοιχα. Αυτές εφαρμόζονται και στην MySQL και στην PostgreSQL αντίστοιχα αλλά με διαφορετικούς μηχανισμούς. Στην MySQL χρησιμοποιούνται τα binary logs , ενώ στην PostgreSQL χρησιμοποιούνται τα WAL αρχεία. Παρακάτω θα αναλυθούν λεπτομερώς τα 4 είδη των backup και για τους 2 τύπους βάσεων δεδομένων.[10][11]

## FULL BACKUP - Πλήρες Αντίγραφο Ασφαλείας

Το **Full Backup** είναι η βασικότερη και πιό ολοκληρωμένη μέθοδος δημιουργίας αντιγράφων ασφαλείας ανάμεσα σε όλα τα είδη των backups. Αποθηκεύει ολόκληρη τη βάση δεδομένων, περιλαμβάνοντας όλα τα δεδομένα αλλά και τη δομή της. Αυτή η μέθοδος είναι η πιο αξιόπιστη, καθώς παρέχει ένα πλήρες αντίγραφο της βάσης, το οποίο μπορεί να αποκατασταθεί εύκολα χωρίς να απαιτεί την χρήση ή τον συνδυασμό προηγουμένων αντιγράφων ασφαλείας. Παρόλα αυτά, καταναλώνει περισσότερο χρόνο και χώρο αποθήκευσης σε σχέση με άλλες μεθόδους, όπως το **incremental**. Συνήθως χρησιμοποιείται στις περιπτώσεις που απαιτείται πλήρης αποκατάσταση και σταθερότητα των δεδομένων των βάσεων. Αυτή η μέθοδος υποστηρίζεται και στην MySQL και στην PostgreSQL.

### Δημιουργία full backup με την χρήση MySQL

Στην **MySQL** [12] , το πλήρες αντίγραφο βάσης δεδομένων μπορεί να δημιουργηθεί με την εντολή mysqldump.



Αν ο χρήστης επιθυμεί να δημιουργήσει πλήρη αντίγραφα όλων των βάσεων δεδομένων, τότε συντάσει την εντολή με αυτόν τον τρόπο:



Στην περίπτωση που ο χρήστης επιθυμεί να δημιουργήσει αντίγραφα για συγκεκριμένες βάσεις δεδομένων, τότε συντάζει την εντολή με την χρήση της επιλογής --databases στην γραμμή εντολών.



[14] Με την την χρήση της mysqldump εντολής η διαδικασία δημιουργίας ενός πλήρες αντιγράφου ασφαλείας είναι ευκολότερη και απλούστερη, λόγω της δυνατότητα αποθήκευσης σε ένα απλό αρχείο sql, προσφέροντας έτσι την δυνατότητα επαναφοράς ολόκληρης της βάσης σε ένα συγκεκριμένο χρονικό σημείο σε περίπτωση απώλειας ή καταστροφής. Έτσι επιτυγχάνεται η καλύτερη δυνατή προστασία και πλήρη αποκατάσταση των δεδομένων, επειδή η καταγραφή αυτή περιλαμβάνει τα πάντα, συμπεριλαμβανομένων όλων των αρχείων δεδομένων, των αρχείων διαμόρφωσης καθώς και των δυαδικών αρχείων καταγραφής.

Ωστόσο υπάρχουν και μειονεκτήματα, με το μεγαλύτερο από αυτά να είναι ο μεγάλος χρόνος της δημιουργίας του πλήρες αντιγράφου, ειδικότερα για μεγάλες βάσεις δεδομένων με μεγάλο αριθμό δεδομένων. Ταυτόχρονα επηρεάζεται και ο αποθηκευτικός χώρος σημαντικά και δεν είναι ιδανικό για συστήματα που απαιτούν συχνές αποθηκεύσεις, με την επιβάρυνση των πόρων του συστήματος να αποτελεί άλλο ένα σημαντικό μειονέκτημα.

### Δημιουργία full backup με την χρήση PostgreSQL

Στην **PostgreSQL** αντίστοιχα [13], χρησιμοποιείται η εντολή pg\_dump και συντάσσεται ως εξής:



Έπειτα για την επαναφορά του αρχείου που δημιουργείται με την pg\_dump, χρησιμοποιείται η εντολή psql:

,

όπου στην παραπάνω γραμμή εντολών το dumpfile είναι το αρχείο που δημιουργήθηκε από την pg\_dump.

Πριν την αποκατάσταση ενός πλήρες αντιγράφου ασφαλείας, ο χρήστης θα πρέπει να έχει όλα τα δικαιώματα για όλα τα δεδομένα και τα αντικείμενα που περιλαμβάνονται στο αντίγραφο ασφαλείας. Εάν αυτά δεν υπάρχουν, η πλήρη επαναφορά της βάσης θα αποτύχει.

Μπορεί να αποθηκεύσει τόσο την δομή της βάσης όσο και τα δεδομένα της σε διάφορες μορφές αρχείων, όπως αρχεία sql, αρχεία tar η και σε custom format. Ένα από τα μεγαλύτερα πλεονεκτήματα του είναι ότι μπορεί να εκτελεστεί από οποιαδήποτε απομακρυσμένο σύστημα έχει πρόσβαση στην εκάστοτε βάση δεδομένων, χωρίς να διακόπτει τις λειτουργίες της. Επιπροσθέτως, τα αρχεία που δημιουργούνται είναι συμβατά με νεότερες εκδόσεις PostgreSQL, διευκολύνοντας έτσι τόσο την αναβάθμιση των δεδομένων όσο και την μεταφορά τους μεταξύ διαφορετικών αρχιτεκτονικών συστημάτων.

Ωστόσο η χρήση της pg\_dump επιφέρει και αυτή μειονεκτήματα, από τον ιδιαίτερα αυξημένο χρόνο που απαιτείται για την αποκατάσταση μεγάλων βάσεων δεδομένων μέχρι και του συνόλου των ειδικών δικαιωμάτων που απαιτούνται για την πλήρη εξαγωγή όλων των δεδομένων. Αν ο χρήστης δεν έχει τις απαραίτητες άδειες (SUPERUSER), περιορίζεται στην δημιουργία backups μόνο συγκεκριμένων κομματιών της εκάστοτε βάσης, με την χρήση των επιλογών στην γραμμή εντολών όπως η “-n schema” ή “-t table”. Επιπλέον κάποιες λειτουργίες που απαιτούν αποκλειστική πρόσβαση του χρήστη στην βάση δεδομένων, όπως για παράδειγμα συγκεκριμένες ALTER TABLE εντολές, μπορεί να επηρεάσουν σημαντικά την όλη διαδικασία.

## INCREMENTAL BACKUP - Επαυξητικό Αντίγραφο Ασφαλείας

### 

To **Incremental Backup** είναι μια μέθοδος δημιουργίας αντιγράφων ασφαλείας που αποθηκεύει μόνο τις αλλαγές που έχουν γίνει μετά το τελευταίο backup, περιορίζοντας σημαντικά την αποθήκευση μεγάλου όγκου δεδομένων. Σε αντίθεση με το full backup που περιλαμβάνει ολόκληρη την βάση, το incremental backup καταγράφει μόνο τις νέες ή τροποποιημένες εγγραφές, επιταχύνοντας την διαδικασία και ταυτόχρονα εξοικονομώντας πόρους, κάτι που το καθιστά την ιδανικότερα επιλογή για μεγάλες βάσεις δεδομένων.

Ωστόσο η διαδικασία αποκατάστασης αυτόματα γίνεται πιο περίπλοκη, καθώς απαιτείται η χρήση και ο συνδυασμός πολλών αντιγράφων ασφαλείας για επιτυχή αποκατάσταση όλων των δεδομένων. Αυτή η μέθοδος υποστηρίζεται και στην MySQL και στην PostgreSQL.

### Δημιουργία incremental backup με την χρήση MySQL

Μέσω της χρήσης των binary logs επιτυγχάνεται η δημιουργία ενός incremental backup στη MySQL. Αυτό συμβαίνει επειδή τα logs καταγράφουν όλες τις αλλαγές που πραγματοποιούνται στην εκάστοτε βάση δεδομένων. Αυτή η μέθοδος δημιουργίας αντιγράφων ασφαλείας επιτρέπει την αποθήκευση μόνο των διαφοροποιήσεων από το τελευταίο οποιουδήποτε είδους backup που πραγματοποιήθηκε, μειώνοντας έτσι σημαντικά τον χρόνο και τον αποθηκευτικό χώρο που απαιτείται σε σύγκριση με ένα πλήρες backup.

Προκειμένου να μπορούν να πραγματοποιηθούν τα incremental backups θα πρέπει τα binary logs να είναι ενεργοποιημένα στον MySQL server, ρυθμίζοντας την παράμετρο *log\_bin* στο αρχείο ρυθμίσεων.

Έπειτα εφόσον υπάρχει το πλήρες backup αρχείο, χρησιμοποιούμε την εντολή mysqlbinlog προκειμένου να εξαχθούν όλες οι αλλαγές που καταγράφηκαν στα binary logs μετά το τελευταίο backup ως εξής:



Μετά την επιτυχή εξαγωγή των αλλαγών, με την χρήση της mysql κάνουμε επαναφορά του incremental backup ως εξής:

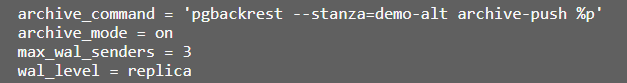


Ένα μεγάλο πλεονέκτημα των επαυξητικών αντιγράφων ασφαλείας είναι η σημαντική μείωση του χρόνου καθώς και του χώρου αποθήκευσης που απαιτούνται, λόγω του ότι αποθηκεύονται μόνο οι αλλαγές από το τελευταίο backup. Ωστόσο για την σωστή επαναφορά του backup απαιτείται η εφαρμογή του πλήρους backup και όλων των επακόλουθων incremental backups με την σωστή σειρά. Ιδιαίτερη προσοχή πρέπει να δίνουμε και στην αποθήκευση και διαχείριση των binary logs προκειμένου να διασφαλιστεί η ακεραιότητα των δεδομένων κατά την αποκατάσταση.[16]

### Δημιουργία incremental backup με την χρήση PostgreSQL

Στην PostgreSQL η διαδικασία δημιουργίας incremental backups είναι λίγο διαφορετική , καθώς εδώ χρησιμοποιούμε το Write-Ahead Logging (WAL), το οποίο καταγράφει όλες τις αλλαγές που γίνονται στην εκάστοτε βάση δεδομένων. Επειδή η εντολή pg\_dump, που είναι η βασική εντολή στην PostgreSQL για την δημιουργία αντιγράφων ασφαλείας, δεν υποστηρίζει incremental backups, χρησιμοποιούμε εργαλεία όπως το pg\_basebackup, pg\_restore και το WAL archiving.

[17]Προκειμένου να δημιουργηθούν incremental backups με WAL , πρέπει πρώτα να ενεργοποιηθεί η καταγραφή των WAL logs με την προσθήκη στο αρχείο postgresql.conf των εξής γραμμών:



Έτσι αποθηκεύονται όλα τα WAL αρχεία στον φάκελο /var/lib/postgresql/wal\_archive/ .

Τα WAL αρχεία που είχαν καταγραφεί μετά το πλήρες backup μπορουν να επαναφερθούν με την χρήση της pg\_restore ως εξής:



Με την χρήση αυτής της μεθόδου δημιουργίας επαυξητικών αντιγράφων ασφαλείας επιτυγχάνεται αυξημένη αποδοτικότητα και ελαχιστοποίηση του αποθηκευτικού χώρου. Παρόλα αυτά υπάρχει και αυξημένη πολυπλοκότητα διαχείρισης των WAL logs και πρέπει να υπάρχει προσεκτικός συγχρονισμός με το πλήρες backup. [18]

## DIFFERENTIAL BACKUP - Διαφορικό Αντίγραφο Ασφαλείας

[19]Το differential backup είναι μια μέθοδος δημιουργία αντιγράφων ασφαλείας που αποθηκεύει όλες τις αλλαγές που έχουν γίνει από το **τελευταίο πλήρες αντίγραφο**, σε αντίθεση με το incremental, οπου εκεί κάθε αντίγραφο περιλαμβάνει μόνο τις αλλαγές από το τελευταίο backup (είτε πλήρες είτε incremental). Αυτή η μέθοδος μειώνει τον χρόνο επαναφοράς, καθώς απαιτεί μόνο το αρχικό πλήρες backup και το πιο πρόσφατο differential backup. Ωστόσο, όσο αυξάνεται ο αριθμός των differential backups που δημιουργούνται τόσο αυξάνεται και το μέγεθος του καθενός, καθώς περιλαμβάνουν όλες τις αλλαγές από το τελευταίο πλήρες backup.

### Δημιουργία differential backup με την χρήση MySQL

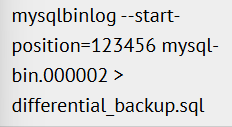
Στην MySQL, η δημιουργία differential backup επιτυγχάνεται μέσω της χρήσης των binary logs, στα οποία καταγράφονται όλες οι τροποποιήσεις στην εκάστοτε βάση δεδομένων. Αυτά τα αρχεία περιέχουν όλες τις απαιτούμενες, για την εύκολη αποκατάσταση δεδομένων σε συγκεκριμένο χρονικό σημείο, αλλαγές που έγιναν μετά το τελευταίο πλήρες αντίγραφο ασφαλείας.

[19]Πρώτα πρέπει να έχει δημιουργηθεί ένα πλήρες αντίγραφο ασφαλείας και να ξεκινήσει ένα νέο binary log με τον παρακάτω τρόπο:



, όπου στην παραπάνω εντολή το --flush-logs αναγκάζει τον server να δημιουργήσει ένα νέο binary log, ενώ η --master-data=2 καταγράφει τη θέση του δυαδικού αρχείου καταγραφής στο αρχείο αντιγράφων ασφαλείας.

Μετά την δημιουργία του πλήρες backup, τα binary logs καταγράφουν κανονικά τις αλλαγές. Προκειμένου να δημιουργηθεί ένα διαφορικό αντίγραφο ασφαλείας, απαιτείται η χρήση της εντολής mysqlbinlog για να εξαχθούν οι αλλαγές από το τελευταίο πλήρες backup με την σύνταξη της εντολής ως εξής:



,όπου στην θέση του 123456 τοποθετούμε την θέση του τελευταίου backup και στην θέση του mysql-bin.000002 τοποθετούμε το όνομα του αντίστοιχου binary log.

### Δημιουργία differential backup με την χρήση PostgreSQL

Στην PostgreSQL, η δημιουργία differential backup έχει αρκετές ομοιότητες με την διαδικασία δημιουργίας incremental, με την διαφορά ότι εδώ θέλουμε την καταγραφή όλων των αλλαγών που έγιναν από το τελευταίο πλήρες backup, επιτυγχάνοντας έτσι ταχύτερη επαναφορά δεδομένων.

Εφόσον έχουμε ενεργοποιήσει το WAL Archiving με τον ίδιο τρόπο όπως είχαμε προαναφέρει και στο [incremental](#_heading=h.eusj6cfdzatf) κομμάτι της postgreSQL , και έχουμε συντάξει την εντολή δημιουργίας πλήρους backup [20] ως εξής:



, προχωράμε στη δημιουργία differential backup, αντιγράφοντας μόνο τα νέα WAL αρχεία χρησιμοποιώντας την εντολή rsync ως εξής:



Για την αποκατάσταση των δεδομένων χρησιμοποιούμε το τελευταίο πλήρες backup και τα διαφορικά WAL αρχεία.

Συνοψίζοντας το differential backup μειώνει αρκετά τον χρόνο επαναφοράς δεδομένων, ωστόσο μπορεί να χρειάζεται περισσότερο αποθηκευτικό χώρο λόγω της συσσώρευσης αρχείων.

## SNAPSHOT BACKUP - Αντίγραφο Ασφαλείας Στιγμιοτύπου

To snapshot backup είναι μια μέθοδος δημιουργίας αντιγράφων ασφαλείας που καταγράφει την κατάσταση της εκάστοτε βάσης δεδομένων σε μια συγκεκριμένη χρονική στιγμή. Αντί να αντιγράφονται όλα τα δεδομένα, δημιουργείται ένα στιγμιότυπο του αποθηκευτικού χώρου, καταγράφοντας τις αλλαγές μετά το αρχικό snapshot, επιτυγχάνοντας έτσι ταχύτερη δημιουργία backup και αποκατάσταση των δεδομένων.

Το μεγάλο πλεονέκτημα του αντιγράφου ασφαλείας στιγμιότυπου είναι η αυξημένη ταχύτητα αλλα και αποδοτικότητα, καθώς δεν απαιτεί πλήρη αντιγραφή των δεδομένων. Παρόλα αυτά, οι αποδοτικότητα τους εξαρτάται από το εκάστοτε σύστημα αποθήκευσης και μπορεί να χρειαστεί περισσότερο αποθηκευτικό χώρο αν υπάρχουν πολλές αλλαγές στα δεδομένα.

Τα snapshot backups αποθηκεύει τα μεταδεδομένα που σχετίζονται με κάθε μπλοκ δεδομένων και κάθε φορά που υπάρχει μια αλλαγή, τα νέα μεταδεδομένα καταγράφονται, επιτρέποντας ένα αρχείο καταγραφής αλλαγών και την ανάπτυξη αντιγράφων ασφαλείας σε πραγματικό χρόνο, όταν αποκαλύπτεται ένα σφάλμα ή μια παραβίαση δεδομένων[21].

### Δημιουργία snapshot backup με την χρήση MySQL

[22]Προκειμένου να δημιουργήσουμε snapshot backup στη MySQL, χρησιμοποιούμε LVM (Logical Volume Manager) snapshots. Πρέπει ωστόσο να σημειωθεί ότι είναι σημαντικό να κρατηθεί κάποιο μέρος του φυσικού χώρου του δίσκου ως μη διατιθέμενο, καθώς ο μη διατεθειμένος χώρος είναι αυτός που επιτρέπει την δημιουργία snapshot.

Η πιο αποδοτική πρακτική είναι η διάθεση που αποθηκευτικού χώρου που απαιτείται με τα σημερινά δεδομένα συν 10% για ανάπτυξη. Συγκριτικά με τις παραδοσιακές κατατμήσεις δίσκου, η επέκταση ενός τόμου LVM είναι γρηγορότερη, ευκολότερη και δεν απαιτεί χρόνο διακοπής λειτουργίας.

Ωστόσο αρκετοί πάροχοι cloud δεν υποστηρίζουν την τεχνολογία LVM. Παρόλα αυτά για όλους εκείνους τους διακομιστές που μπορούν να χρησιμοποιήσουν LVM, η λειτουργία αντιγράφου ασφαλείας στιγμιοτύπου μπορεί να χρησιμοποιηθεί για να ληφθεί ένα εφεδρικό αντίγραφο ασφαλείας MySQL της εκάστοτε βάσης σε συγκεκριμένο χρονικό σημείο με ελάχιστο κλείδωμα πινάκων.

Η χρονική διάρκεια όλης αυτής της διαδικασίας εξαρτάται από την ποσότητα των δεδομένων που πρέπει να εγγραφούν στον δίσκο, αλλά σε γενικές γραμμές είναι πολύ σύντομη.

Με την χρήση της εντολής:



κλειδώνουμε όλους τους πίνακες, αποτρέποντας τις εγγραφές και τις αλλαγές στα δεδομένα αλλά επιτρέποντας την ανάγνωση. Εκτελείται πριν την δημιουργία ενός snapshot backup.

Στην συνέχεια δημιουργούμε το snapshot με την χρήση LVM, διαδικασία η οποία είναι σχεδόν στιγμιαία καθώς χρησιμοποιείται μια copy-on-write (COW) snapshot μέθοδος:



Τώρα που το snapshot backup έχει επιτυχώς δημιουργηθεί, ξεκλειδώνουμε τους πίνακες με την:

.

### Δημιουργία snapshot backup με την χρήση PostgreSQL

[23]Προκειμένου να δημιουργήσουμε ένα snapshot backup σε PostgreSQL βάση δεδομένων, χρησιμοποιούμε την εντολή pg\_dump. Αυτή η εντολή δημιουργεί συνεπή στιγμιότυπα της βάσης, ακόμα και αν αυτή χρησιμοποιείται εκείνη την στιγμή. Η διαφορά της με την MySQL είναι ότι αυτή δεν εμποδίζει άλλους χρήστες να διαβάσουν ή να γράψουν στη βάση.

Ωστόσο πρέπει να υπάρχουν δικαιώματα SELECT στην βάση δεδομένων που θέλουμε να δημιουργηθεί το snapshot backup. Η δημιουργία του σε μορφή απλού κειμένου μπορεί να αποκατασταθεί με την χρήση της εντολής psql. Τα snapshot backups που δημιουργούνται σε μορφή αρχείου αποκαθίστανται αντίστοιχα με την εντολή pg\_restore.

Εφόσον επιβεβαιώσουμε ότι το σύστημα αρχείων μας υποστηρίζει τα snapshots, όπως το LVM (Logical Volume Manager) ή το ZFS, εκτελούμε την παρακάτω εντολή:



, προκειμένου να ενημερώσουμε την PostgreSQL ότι ξεκινά ένα αντίγραφο ασφαλείας, επιτρέποντας την δημιουργία ενός αντιγράφου ασφαλείας στιγμιότυπου.

Στην συνέχεια χρησιμοποιούμε το αντίστοιχο εργαλείο που υποστηρίζει το σύστημα μας για την δημιουργία του στιγμιοτύπου. Για παράδειγμα με το LVM εκτελούμε την εντολή:

.

Μετά την επιτυχή δημιουργία του αντιγράφου ασφαλείας στιγμιοτύπου, εκτελούμε την εντολή:



, ενημερώνοντας έτσι την PostgreSQL για το τέλος του αντιγράφου ασφαλείας και διασφαλίζοντας ότι όλα τα απαραίτητα αρχεία καταγραφής έχουν αποθηκευτεί.

# Τεχνολογίες

## HTML

[24]Η HTML (HyperText Markup Language) είναι η θεμελιώδης και η πιο διαδεδομένη παγκοσμίως γλώσσα περιγραφής ιδιοτήτων των στοιχείων μιας ιστοσελίδας, που χρησιμοποιείται για την δημιουργία ιστοσελίδων και web εφαρμογών. Χρησιμοποιήθηκε στην πτυχιακή μου εργασία προκειμένου να σχεδιαστεί η διεπαφή του χρήστη, παρέχοντας τη δομή και την βάση για την απόδοση του περιεχομένου της Server Data Backup εφαρμογής μου.

Ο όρος “HyperText” , που μεταφράζεται ως “υπερκείμενο” , αναφέρεται σε συνδέσμους που συνδέουν ιστοσελίδες μεταξύ τους, είτε εντός του ίδιου δικτυακού τόπου είτε μεταξύ δικτυακών τόπων διαφορετικών τοποθεσιών. Οι σύνδεσμοι αυτοί είναι μια θεμελιώδης πτυχή του Παγκόσμιου Ιστού. Με το ανέβασμα οποιουδήποτε είδους περιεχομένου και με την σύνδεση του με ιστοσελίδες που έχουν δημιουργηθεί από άλλους ανθρώπους, γινόμαστε ενεργοί συμμετέχοντες στον Παγκόσμιο Ιστό.

## Θετικά χαρακτηριστικά της HTML

[25]Ένα από τα βασικότερα πλεονεκτήματα της HTML είναι η απλότητα της, καθώς επιτρέπει τη γρήγορη ανάπτυξη αλλά και διαμόρφωση ιστοσελίδων. Είναι μια ευρέως υποστηριζόμενη γλώσσα περιγραφής ιδιοτήτων των στοιχείων από τα οποία αποτελείται μια ιστοσελίδα, από όλους τους σύγχρονους browsers, εξασφαλίζοντας την συμβατότητα μεταξύ διαφορετικών πλατφορμών.

Η HTML είναι εύκολη στην εκμάθηση και ο συνδυασμός της με τεχνολογίες JAVASCRIPT και CSS, διευκολύνει και βελτιώνει σε τεράστιο βαθμό την εμπειρία χρήστη.

Είναι SEO-Friendly[26], γεγονός που διευκολύνει την ευρετηρίαση των ιστοσελίδων από πολυάριθμες μηχανές αναζήτησης. Αυτό το είδος περιεχομένου έχει δημιουργηθεί με τέτοιο τρόπο που βοηθά τις μηχανές αναζήτησης να το κατατάξουν ψηλά. Αν και λανθασμένα κυκλοφορεί η αντίληψη ότι πρέπει να είναι γεμάτο με λέξεις κλειδιά, στην πραγματικότητα ο στόχος είναι να καθοδηγήσει τις μηχανές αναζήτησης να βρουν,να κατανοήσουν και εν τέλη να συνδέσουν το περιεχόμενο μας με το θέμα που προσπαθούμε να καλύψουμε.

## Αρνητικά χαρακτηριστικά της HTML

[27]Παρότι η HTML είναι απαραίτητη για την δομή των ιστοσελίδων, έχει περιορισμένες επιλογές σχεδίασης. Δεν μπορεί από μόνη της να παρέχει προηγμένα στυλιστικά χαρακτηριστικά, καθώς το σχεδιαστικό κομμάτι αναλαμβάνεται κυρίως από την CSS. Χωρίς την χρήση της CSS, οι ιστοσελίδες μπορεί να είναι ανταποκρίσιμες και να εκτελούν τα εκάστοτε αιτήματα του χρήστη αλλά φαίνονται απλές και μη στυλιστικές. Επίσης η HTML δεν υποστηρίζει δυναμικά animations ή αλληλεπιδράσεις, κάτι που απαιτεί ξανά την χρήση άλλης τεχνολογίας, στην περίπτωση αυτή της JAVASCRIPT.

Ο σχεδιασμός με HTML είναι λιγότερο ευέλικτος σε σύγκριση με άλλες τεχνολογίες, όπως τα frameworks (π.χ. React, Angular), που προσφέρουν πιο προσαρμόσιμες λύσεις. Γενικότερα για σύνθετες διεπαφές χρήστη, η HTML πρέπει να συνδυάζεται με άλλες τεχνολογίες, γεγονός που αυξάνει την δυσκολία ανάπτυξης.

Ένα ακόμη σημαντικό μειονέκτημα της HTML είναι η μη υποστήριξη offline περιήγησης. Οι ιστοσελίδες που δημιουργούνται με αυτήν απαιτούν να υπάρχει σύνδεση στο διαδίκτυο προκειμένου να μπορεί να φορτωθεί το περιεχόμενο. Όταν ένας χρήστης προσπαθεί να επισκεφθεί μια HTML σελίδα χωρίς σύνδεση, το πρόγραμμα περιήγησης δεν μπορεί να ανακτήσει αρχεία από τον διακομιστή, με αποτέλεσμα η σελίδα να μην είναι διαθέσιμη. Για την υποστήριξη offline browsing, απαιτείται η χρήση τεχνολογιών όπως το Service Workers με το Cache API.

[28]Οι Service Workers ουσιαστικά είναι σκριπτάκια που τρέχουν στο background ενός browser και επιτρέπουν την αποθήκευση και διαχείριση δεδομένων για offline χρήση. Όταν ένας χρήστης επισκέπτεται ξανά τη σελίδα, ο Service Worker μπορεί να εξυπηρετήσει τα δεδομένα από την μνήμη cache, βελτιώνοντας την ταχύτητα και την διαθεσιμότητα του διαδικτυακού περιεχομένου.

Χωρίς αυτές τις τεχνικές και με την έλλειψη σύνδεσης στο διαδίκτυο, οι HTML σελίδες δεν μπορούν να λειτουργήσουν, καθιστώντας την εμπειρία του χρήστη περιορισμένη σε περιβάλλοντα με χαμηλή ή μηδενική συνδεσιμότητα. Έτσι καταλήγουμε πάλι στο συμπέρασμα ότι η HTML πρέπει να συνδυάζεται με άλλες τεχνολογίες για την παροχή πιο αξιόπιστου και δυναμικού περιεχομένου.

## Συμπέρασμα και αιτιολόγηση επιλογής της HTML

Λαμβάνοντας υπόψην όλα τα παραπάνω, η HTML αποδεικνύεται ότι ήταν η καταλληλότερη επιλογή καθώς συνδυάζει απλότητα, ευελιξία και συμβατότητα, χαρακτηριστικά που έκαναν δυνατή την δημιουργία αλλά και την λειτουργικότητα της web εφαρμογής μου. Παρότι απαιτεί την χρήση επιπλέον τεχνολογιών για δυναμικότερο περιεχόμενο, παραμένει απαραίτητη στην δημιουργία κάθε web εφαρμογής.

Ένας ακόμη σημαντικός λόγος που την επέλεξα για την πτυχιακή μου εργασία ήταν επειδή επιτρέπει την εύκολη ενσωμάτωση με τις υπόλοιπες τεχνολογίες που χρησιμοποίησα (PHP, SQL, CSS). Αυτή η ευελιξία της με βοήθησε ιδιαίτερα στο να χτίσω την εφαρμογή μου και να κάνω ευκολότερα προσιτή και περισσότερο εύχρηστη την διεπαφή του χρήστη.

## CSS

Η δεύτερη και εξίσου σημαντική τεχνολογία που χρησιμοποίησα για την βελτίωση της διάταξης αλλά και αισθητικής της web εφαρμογής μου είναι η CSS (Cascading Style Sheets). Πρόκειται για ακόμα μία θεμελιώδη τεχνολογία του Web Development, παρέχοντας πολυάριθμες τεχνικές μορφοποίησης των HTML στοιχείων και διαμόρφωσης responsive σχεδίων που προσαρμόζονται σε διάφορες συσκευές.

[29]Με την χρήση της CSS, εξοικονομείται πολύς χρόνος και δουλειά. Μπορεί να ελέγξει την διάταξη πολλών ιστοσελίδων ταυτόχρονα. Υπάρχουν τρεις τρόποι με τους οποίους μπορεί να προστεθεί CSS σε στοιχεία HTML.

Ο πρώτος τρόπος είναι ‘INLINE’ , που μεταφράζεται ως ‘ΕΝΤΟΣ’, όπου το χαρακτηριστικό στυλ χρησιμοποιείται μέσα στα στοιχεία HTML. Ο δεύτερος τρόπος είναι μέσα στο αρχείο html όπου το στοιχείο <style> χρησιμοποιείται στην ενότητα <head>. Ο τρίτος και ο πιο διαδεδομένος τρόπος με τον οποίο χρησιμοποιείται η css είναι σε εξωτερικά αρχεία .css, όπου εκεί αποθηκεύονται όλες οι δαιμορφώσεις τον στοιχείων. Ένα εξωτερικό αρχείο .css καθορίζει το στυλ για πολλαπλές σελίδες HTML. Μπορεί με οποιαδήποτε αλλαγή πραγματοποιηθεί μέσα σε αυτό το αρχείο να αλλαχτεί η εμφάνιση ολόκληρου του ιστότοπου. Ωστόσο για να μπορεί να χρησιμοποιηθεί ένα ή περισσότερα εξωτερικά φύλλα στυλ, θα πρέπει να προστεθούν οι αντίστοιχοι σύνδεσμοι προς αυτό ή αυτά μέσα στο στοιχείο της HTML.

## Θετικά χαρακτηριστικά της CSS

[30]Ένα από τα μεγαλύτερα πλεονεκτήματα της χρήσης CSS είναι ότι εξοικονομεί πολύτιμο χρόνο, καθώς προσαρμόζει την διάταξη πολλών ιστοσελίδων ταυτόχρονα. Όλα τα εξωτερικά stylesheets αποθηκεύονται σε αρχεία .css, τα οποία μπορούν να συνδεθούν με πολλαπλές σελίδες HTML, διευκολύνοντας σε μεγάλο βαθμό τόσο την συντήρηση όσο και την ενημέρωση του σχεδιασμού.

Η χρήση της CSS συμβάλλει επίσης στην βελτίωση της ταχύτητας με την οποία φορτώνει μια ιστοσελίδα, εφόσον μειώνεται σημαντικά η επανάληψη κώδικα καθώς επιτρέπει την αποθήκευση των διαφόρων στυλ που χρησιμοποιεί στην μνήμη cache του browser.

Παράλληλα υποστηρίζει τη δημιουργία responsive σχεδίων, εξασφαλίζοντας έτσι ότι όλες οι ιστοσελίδες θα εμφανίζονται σωστά σε διαφορετικές συσκευές με διαφορετικές διαστάσεις, όπως υπολογιστές, tablets και κινητά τηλέφωνα.

## Αρνητικά χαρακτηριστικά της CSS

[31]Παρά τα πολυάριθμα οφέλη της χρήσης CSS, υπάρχουν αρκετά μειονεκτήματα που μπορεί να επηρεάσουν σε μεγάλο βαθμό την ανάπτυξη των ιστοσελίδων. Ένα από αυτά είναι η ασυνέπεια που υπάρχει ανάμεσα διαφορετικών browser, καθώς οι εκδόσεις της CSS ερμηνεύονται με διαφορετικό τρόπο σε προγράμματα περιήγησης (Chrome, Firefox, Safari κτλ.). Αυτό σημαίνει ότι οι προγραμματιστές θα πρέπει να κάνουν επιπλέον δοκιμές και πολύπλοκες προσαρμογές προκειμένου να επιτευχθεί η εκάστοτε συμβατότητα.

Ένα άλλο σημαντικό μειονέκτημα είναι η πολυπλοκότητα της διαχείρισης CSS αρχείων μεγάλου μεγέθους. Όσο αυξάνεται το μέγεθος ενός project, τόσο αυξάνεται και η δυσκολία και πολυπλοκότητα της συντήρησης και της οργάνωσης του κώδικα, ειδικότερα άμα δεν υιοθετείται η χρήση preprocessor CSS εργαλείων (SASS, LESS). Επιπλέον, η CSS δεν διαθέτει ενσωματωμένο μηχανισμό για μεταβλητές και επαναχρησιμοποιήσιμο κώδικα αντίστοιχα, κάνοντας έτσι την ανάπτυξη πιο απαιτητική και χρονοβόρα.

Τελευταίο και άξιο αναφοράς είναι η διευκρίνιση πως όταν αναφερόμαστε στην CSS, δεν αναφερόμαστε σε γλώσσα προγραμματισμού, κάτι που περιορίζει τις δυνατότητες για δυναμική διαχείριση περιεχομένου. Προκειμένου να αντιμετωπιστεί αυτός ο περιορισμός, χρησιμοποιούνται frameworks (όπως το React) αλλά και η Javascript.

## PHP

Σημαντικός παράγοντας για την ανάπτυξη της web εφαρμογής μου και επίτευξης του ζητουμένου της πτυχιακής μου εργασίας ήταν η χρήση της PHP τεχνολογίας, μία από τις πιο διαδεδομένες γλώσσες προγραμματισμού για δυναμικές ιστοσελίδες. [32]Πρόκειται για μια γλώσσα που εκτελείται στον server και επιτρέπει την διαχείριση δεδομένων, την αλληλεπίδραση με βάσεις δεδομένων(που ήταν και το κύριο ζητούμενο της πτυχιακής) αλλά και την δημιουργία δυναμικού περιεχομένου.

[34]Η HTML μπορεί να ενσωματώσει την PHP και με αυτό τον τρόπο καθίστανται δυνατή η επικοινωνία με οποιαδήποτε βάση δεδομένων και η εκτέλεση ποικιλίας εργασιών. Αυτό σημαίνει ότι τόσο η είσοδος χρήστη όσο και άλλες μεταβλητές μπορούν να χρησιμοποιηθούν για την τροποποίηση ιστοσελίδων. Επίσης η php χρησιμοποιείται για αποστολή και λήψη cookies, για συλλογή μορφοποιημένων δεδομένων αλλά και για ενημέρωση πληροφοριών βάσεων δεδομένων.

“PHP: Hypertext Preprocessor” είναι το ακρωνύμιο της PHP. Επειδή αναφερόμαστε σε μια γλώσσα αποτελούμενη από ένα σύνολο ενεργειών που πραγματοποιούνται από την πλευρά του διακομιστή, ο κώδικας εκτελείται στον διακομιστή πριν το πρόγραμμα περιήγησης του χρήστη λάβει το περιεχόμενο της ιστοσελίδας.

## Θετικά χαρακτηριστικά της PHP

[34]Ένας από τους κυριότερους λόγους που επέλεξα την php είναι η απλότητα της αλλα και το πόσο εύκολα έμαθα τις τεχνολογίες της και το πόσο βοηθητικές ήταν στη δημιουργία της εφαρμογής μου.Η σύνταξη της είναι παρόμοια με αυτή των γλωσσών προγραμματισμού όπως είναι η C και η Javascript, κατι που με διευκόλυνε σε μεγάλο βαθμό. Επιπλέον το γεγονός ότι είναι ανοιχτού κώδικα την κάνει πιο προσιτή και πιο οικονομική λύση για να αναπτυχθούν web εφαρμογές.

Σημαντικό πλεονέκτημα της χρήσης της είναι η υποστήριξη πολλαπλών βάσεων δεδομένων, όπως MySQL και PostgreSQL (τα 2 είδη βάσεων που χρησιμοποιήθηκαν στην εφαρμογή), κάνοντας ευκολότερη την υλοποίηση της διαχείρισης αντιγράφων ασφαλείας βάσεων δεδομένων. Ο ταχύτερος ρυθμός φόρτωσης της σε σχέση με άλλες γλώσσες προγραμματισμού ακόμα και με σύνδεση σε δίκτυο αργής ταχύτητας την καθιστά εξαιρετική επιλογή.

Βοηθά σημαντικά στην επαναχρησιμοποίηση ενός ισοδύναμου κώδικα, μειώνοντας έτσι την ανάγκη χρήσης μακροσκελή κώδικα και πολύπλοκων δομών του για την εκτέλεση των απαιτούμενων εργασιών της εφαρμογής.

[35]Ταυτόχρονα διαθέτει ενσωματωμένους μηχανισμούς για την ανίχνευση και διαχείριση σφαλμάτων, διευκολύνοντας σημαντικά την επιδιόρθωση τους. Οι κύριοι τύποι σφαλμάτων είναι τα Parse Errors (Συντακτικά Σφάλματα), τα Fatal Errors (Κρίσιμα Σφάλματα) και τα Warnings & Notices (Προειδοποιήσεις και Ειδοποιήσεις). Με την σωστή διαχείριση σφαλμάτων βελτιώνεται σημαντικά η ασφάλεια και η σταθερότητα των εφαρμογών.

## Αρνητικά χαρακτηριστικά της PHP

[36]Παρότι η PHP περικλείεται από αρκετά θετικά χαρακτηριστικά, την συνοδεύουν και μερικά σημαντικά μειονεκτήματα, με το κυριότερο από αυτά να είναι οι κίνδυνοι ασφάλειας.

Λόγω του ότι αναφερόμαστε σε μια ανοιχτού κώδικα τεχνολογία, καθώς και του εύκολα προσβάσιμου αρχείου κειμένου ASCII, οι εφαρμογές PHP είναι αρκετά ευάλωτες σε επιθέσεις τύπου SQL Injections και cross-site scripting (XSS). Αυτό ‘αναγκάζει’ τους προγραμματιστές να εφαρμόζουν τακτικά βέλτιστες πρακτικές ασφάλειας και να ενημερώνουν τις γνώσεις τους σχετικά με τα εκάστοτε μέτρα ασφαλείας της PHP.

Η περιορισμένη υποστήριξη multithreading είναι επίσης ένα μεγάλο μειονέκτημα της PHP, καθώς το μονονηματικό μοντέλο που ακολουθεί η PHP δεν την καθιστά την καλύτερη επιλογή για πιο περίπλοκες multithreading εφαρμογες. Ενώ η χρήση εξωτερικών βιβλιοθηκών και η ενσωμάτωση με άλλες τεχνολογίες περιορίζουν ως ένα βαθμό το πρόβλημα αυτό, αυτόματα προστίθενται αρκετή πολυπλοκότητα στην διαδικασία ανάπτυξης των εκάστοτε εφαρμογών.

[37]Ενώ η PHP έχει πολλές ενημερώσεις, εξακολουθεί να θεωρείται λιγότερο πλούσια σε χαρακτηριστικά σε σύγκριση με νεότερες γλώσσες προγραμματισμού που έχουν σχεδιαστεί για την ανάπτυξη ιστοσελίδων, όπως για παράδειγμα η JavaScript (Node.js) ή η Python.

## SQL

Η πιο δημοφιλής γλώσσα που χρησιμοποίησα για την διαχείριση βάσεων δεδομένων, που στην περίπτωση της εφαρμογής αυτής κύριο ζητούμενο ήταν η δημιουργία αντιγράφων ασφαλείας βάσεων δεδομένων, είναι η SQL (Structured Query Language). Συγκεκριμένα, επέλεξα να χρησιμοποιήσω MySQL έναντι της PostgreSQL λόγω της εξαιρετικής ευκολίας της στην χρήση και της ευχρηστίας της, χαρακτηριστικά που την καθιστούν ιδανικότερη για την ανάπτυξη της web εφαρμογής μου. Παρότι η PostgreSQL είναι πιο προηγμένη, καθώς υποστηρίζει πιο σύνθετα χαρακτηριστικά και τύπους δεδομένων, η MySQL προσφέρει μια πιο απλοποιημένη εμπειρία, κάτι ιδιαίτερα βοηθητικό στην δημιουργία της web εφαρμογής μου. Η υποστήριξη της από πολλές πλατφόρμες και το γεγονός ότι προσφέρει υψηλή απόδοση για εφαρμογές με υψηλό φόρτο αναγνωστικών λειτουργιών, την έκαναν μονόδρομη επιλογή για αυτό το είδος εφαρμογής.

[38]Γενικότερα, σε μια χρονική περίοδο όπου τα δεδομένα είναι κρίσιμο μέρος πολυάριθμων web και mobile εφαρμογών, χρειάζονται βάσεις δεδομένων και ένας τρόπος για τους χρήστες να μπορούν να διαχειρίζονται αυτά τα δεδομένα. Η SQL είναι η γλώσσα προγραμματισμού που δίνει την δυνατότητα στους προγραμματιστές να διαχειρίζονται αυτά τα δεδομένα.

Ενώ η SQL θεωρείται γλώσσα προγραμματισμού, διαφέρει από τις άλλες γλώσσες προγραμματισμού όπως είναι η C++ και η Java (γλώσσες 3ης γενιάς έναντι της Sql που είναι 4ης). Η σημαντικότερη διαφορά της με τις υπόλοιπες διαδεδομένες γλώσσες προγραμματισμού είναι ότι είναι αρκετά μοναδική, καθώς πρόκειται κυρίως για μια γλώσσα ερωτημάτων που επιτρέπει την αποθήκευση, ανάκτηση αλλά και επεξεργασία δεδομένων σε βάσεις δεδομένων.

## Θετικά χαρακτηριστικά της SQL

[39]Ένα από τα κύρια πλεονεκτήματα της SQL είναι η απλότητα των εντολών ανάκτησης και επεξεργασίας δεδομένων βάσεων που χρησιμοποιεί. Με τον συνδυασμό των εντολών SELECT, INSERT, UPDATE και DELETE, δίνεται η δυνατότητα στους προγραμματιστές να ανασύρουν και να επεξεργαστούν συγκεκριμένα σημεία δεδομένων των εκάστοτε βάσεων. Βοηθάνε στην πραγματοποίηση τροποποιήσεων σε υπάρχουσες εγγραφές καθώς και στην αφαίρεση ανεπιθύμητων δεδομένων. Σε μια εποχή που οι επιχειρήσεις υφίστανται ταχεία αύξηση των δεδομένων, η ικανότητα εύκολης και αποτελεσματικής συσχέτισης των δεδομένων καθίστανται ζωτικής σημασίας πτυχή.

Η ασφάλεια των δεδομένων αποτελεί άλλη μια ύψιστης σημασίας προτεραιότητα στον σύγχρονο ψηφιακό κόσμο, με τις απειλές στον κυβερνοχώρο να αυξάνονται και να εξελίσσονται τακτικά. Οι βάσεις δεδομένων SQL παρέχουν ισχυρούς μηχανισμούς εξασφαλίζοντας όσο το δυνατόν μεγαλύτερη προστασία των ευαίσθητων πληροφοριών. Οι τρόποι με τους οποίους οι βάσεις δεδομένων SQL διασφαλίζουν όλα τα δεδομένα είναι με επιλογές κρυπτογράφησης αλλά και με ελέγχους πρόσβασης. Μόνο με το κατάλληλο κλειδί αποκρυπτογράφησης μπορεί ένας χρήστης να αποκωδικοποιήσει τα δεδομένα αυτά, καθώς η διαδικασία της κρυπτογράφησης τα μετατρέπει σε μη αναγνώσιμες μορφές.

Όσον αφορά τους ελέγχους πρόσβασης, κατηγοριοποιεί τους χρήστες και τους χορηγεί διαφορετικά επίπεδα δικαιωμάτων, εξασφαλίζοντας ότι μόνο εξουσιοδοτημένο προσωπικό μπορεί να έχει πρόσβαση στα εκάστοτε δεδομένα.

Με τον συνδυασμό όλων αυτών των μέτρων ασφαλείας, οι επιχειρήσεις αποκτούν την δυνατότητα οικοδόμησης υψηλής εμπιστοσύνης με τους πελάτες τους, διασφαλίζουν σε μεγάλο βαθμό τις πληροφορίες ιδιοκτησίας και συμμορφώνονται ευκολότερα με τους κανονισμούς προστασίας δεδομένων. Με τις επιθέσεις στον κυβερνοχώρο και τις παραβιάσεις των δεδομένων να αυξάνονται σε μεγάλο βαθμό στην σημερινή εποχή, η SQL προσφέρει μια ασφαλή βάση για την διαχείριση ευαίσθητων και προσωπικών δεδομένων.

## Αρνητικά χαρακτηριστικά της SQL

Παρά τα πολυάριθμα πλεονεκτήματα και οφέλη της SQL στην προστασία και διαχείριση των δεδομένων, παρουσιάζονται και μερικά σημαντικά μειονεκτήματα. [39]

Ένα βασικό ζήτημα είναι οι περιορισμένες δυνατότητες της για οριζόντια κλιμάκωση. Καθώς είναι συνηθισμένο για τις βάσεις των δεδομένων να κλιμακώνονται κάθετα, υπάρχει η απαίτηση για συνεχή αναβάθμιση του υπάρχοντος διακομιστή με περισσότερους πόρους, κάτι που σε περιβάλλοντα με αυξημένες απαιτήσεις δεδομένων, μπορεί να είναι αρκετά δαπανηρό αλλά και περιοριστικό.

Επιπροσθέτως, οι SQL βάσεις δεδομένων είναι σχεδιασμένες με τέτοιο τρόπο που τις καθιστά λιγότερο ευέλικτες για ημιδομημένα ή μη δομημένα δεδομένα, η χρήση των οποίων συνηθίζεται σε σύγχρονες εφαρμογές, λόγω της χρήσης προκαθορισμένων σχημάτων. Αυτή η ανάγκη της για αυστηρή δομή αυξάνει σημαντικά την δυσκολία προσαρμογής της σε μεταβαλλόμενες απαιτήσεις δεδομένων.

[40]Σε θέματα απόδοσης, οι SQL βάσεις δεδομένων μπορεί να παρουσιάσουν μειωμένη ταχύτητα σε εφαρμογές με τεράστιους όγκους δεδομένων ή με υψηλό αριθμό ταυτόχρονων χρηστών. Η ανάγκη τους για πολύπλοκες συνενώσεις (με την χρήση της εντολής JOIN), σε συνδυασμό με την αυστηρή τήρηση των κανόνων ACID, επιβραδύνουν σημαντικά την απόκριση του συστήματος.

Τέλος ένας ακόμα αρνητικός παράγοντας της χρήσης SQL βάσεων είναι το συνολικό κόστος της ιδιοκτησίας τους. Η συντήρηση και η διαχείριση μιας SQL βάσης δεδομένων ειδικότερα σε περιπτώσεις που το σύνολο των δεδομένων είναι υψηλό, απαιτεί πιο ισχυρά υλικά και πιο προηγμένες υποδομές πληροφορικής.

Ο συνδυασμός όλων των παραπάνω αρνητικών παραγόντων καθιστούν τις SQL βάσεις δεδομένων λιγότερο κατάλληλες για εφαρμογές που απαιτούν υψηλή ευελιξία, κλιμάκωση και απόδοση.

## Σχεδίαση και Ανάπτυξη του BUDBServer (Backup Database Server)

## Λειτουργικές Απαιτήσεις

Ο στόχος της πτυχιακής αυτής εργασίας μου ήταν να υλοποιηθεί μια εφαρμογή, πλήρως λειτουργική με όσο το δυνατόν πιο user-friendly interface, η οποία θα δίνει την δυνατότητα στους χρήστες να δημιουργούν και να διαχειρίζονται αντίγραφα ασφαλείας για MySQL και PostgreSQL βάσεις δεδομένων. Η ανάπτυξη αυτής της εφαρμογής βασίστηκε αρκετά σε συγκεκριμένες απαιτήσεις χρηστών (user stories), τις οποίες έλαβα υπόψη για να διασφαλίσω ότι το τελικό αποτέλεσμα καλύπτει τις ουσιαστικές ανάγκες πραγματικών χρηστών.

Ξεκινώντας, η εφαρμογή μου περιλαμβάνει πλήρες σύστημα αυθεντικοποίησης χρήστη, με δυνατότητα εγγραφής (sign up), σύνδεσης (login), καθώς και επαναφοράς κωδικού πρόσβασης (μέσω φόρμας forgot/reset password). Η είσοδος γίνεται αποκλειστικά μέσω email, αυξάνοντας έτσι την ευχρηστία και την ασφάλεια του συστήματος και εξασφαλίζοντας την μοναδικότητα του username του κάθε χρήστη. Για την αποθήκευση των δεδομένων των χρηστών και των αντίστοιχων συνδέσεων τους χρησιμοποιείται μια βάση δεδομένων με την χρήση τριών πινάκων, στους οποίους στηρίζεται η λειτουργικότητα όλης της εφαρμογής.

Η δημιουργία αντιγράφων ασφαλείας όπως προανέφερα καλύβει τόσο MySQL όσο και PostgreSQL servers. Για την επίτευξη αυτής της απαίτησης χρηστών, χρησιμοποιούνται οι αντίστοιχες εντολές backup κάθε συστήματος, οι οποίες εκτελούνται μέσω PHP στο backend κομμάτι. Υποστηρίζεται το βασικό είδος πλήρους αντιγράφου ασφαλείας (FULL BACKUP) , με τα υπόλοιπα 3 είδη (INCREMENTAL, DIFFERENTIAL, SNAPSHOT) να προορίζονται για μελλοντικές επεκτάσεις τις εφαρμογής καθώς υπάρχει ένας μεγάλος περιορισμός δυνατοτήτων με τα δικαιώματα του κάθε server που αποθηκεύει ο κάθε χρήστης καθώς και η έλλειψη ιδιοκτησίας του για κάθε μια από τους αποθηκευμένους του server.

Η διεπαφή της εφαρμογής έχει σχεδιαστεί με μεγαλύτερη προτεραιότητα την φιλικότητα προς τον χρήστη και για αυτό χρησιμοποίησα AJAX για την επικοινωνία μεταξύ Frontend και Backend. Μέσω μιας συλλογής από endpoints επιτυγχάνεται μια γρήγορη και ομαλή αλληλεπίδραση χωρίς να χρειάζεται σε κάθε ενέργεια του χρήστη να πραγματοποιούνται περιττές ανανεώσεις των σελίδων. Ο χρήστης μπορεί να δημιουργεί νέες συνδέσεις σε βάσεις δεδομένων στις οποίες έχει πρόσβαση, να επιλέγει για ποιά σύνδεση θέλει να δημιουργήσει αντίγραφο ασφαλείας και τι είδους, καθώς και να αποθηκεύει τα στοιχεία αυθεντικοποίησης στο προσωπικό του προφίλ κρυπτογραφημένα για μεγαλύτερη ασφάλεια.

Μια σημαντική απαίτηση χρήστη ήταν η σωστή οργάνωση των αντιγράφων ασφαλείας. Αυτό επιτεύχθηκε με την χρήση της εντολής **mkdir**, όπου κατά την εκτέλεση κάθε διαδικασίας δημιουργίας backup δημιουργείται αυτόματα φάκελος στον server, οργανωμένος ανά χρήστη και ανά σύνδεση βάσης δεδομένων, με το όνομα του οποίου να βασίζεται στο email του χρήστη. Επιπλέον, χάρη σε αυτή την οργάνωση, ο χρήστης μπορεί να προβάλει τα αποθηκευμένα του αντίγραφα ασφαλείας, με πληροφορίες για το πότε δημιουργήθηκε και τι είδους είναι αντίστοιχα. Ο χρήστης μπορεί να κατεβάσει τοπικά κάθε αρχείο που έχει δημιουργήσει αλλά και να το διαγράψει σε περίπτωση που το επιθυμεί.

Συνοψίζοντας, κάθε μία από αυτές τις επιλογές τεχνολογιών και εργαλείων έγινε με κεντρικό γνώμονα τις ανάγκες του κάθε συστήματος και των χρηστών του. Αυτό που είχα κυριότερο στόχο ήταν να αποδείξω πως με την χρήση της PHP και κανενός framework μπορεί να υπάρχει μια πλήρως λειτουργική εφαρμογή που να παρέχει μια υψηλής σημασίας υπηρεσία, που είναι η διαχείριση αντιγράφων ασφαλείας βάσεων δεδομένων, με πρακτική αξία στον χώρο της διαχείρισης και προστασίας των δεδομένων στην σύγχρονη αυτή εποχή.

## Αρχιτεκτονική Εφαρμογής

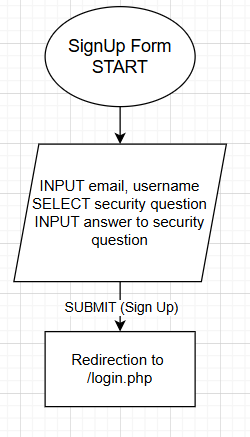
## Flowcharts (Διαγράμματα Ροής)

Παρακάτω παρουσιάζονται τα διαγράμματα ροής της εφαρμογής μου, που αναπαριστούν με απλό και κατανοητό τρόπο πως συνδέονται τα πάντα στην εφαρμογή μου και πως λειτουργεί με τον συνδυασμό τους. Μέσω αυτών φαίνεται η ροή των δεδομένων και τα βασικά βήματα που ακολούθησα για να επιτευχθούν όλες οι λειτουργικές απαιτήσεις της εφαρμογής που προανέφερα στην προηγούμενη ενότητα.

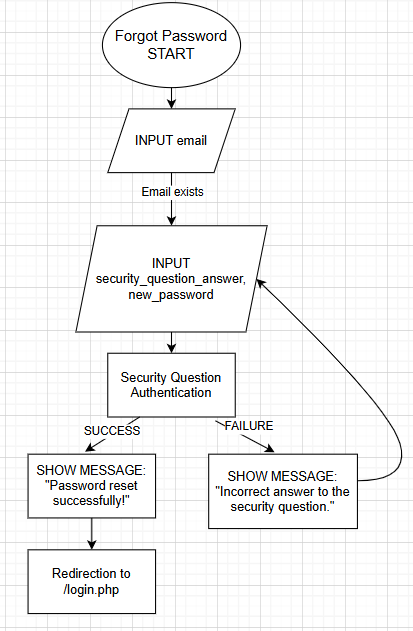
**LOGIN FORM**

# 

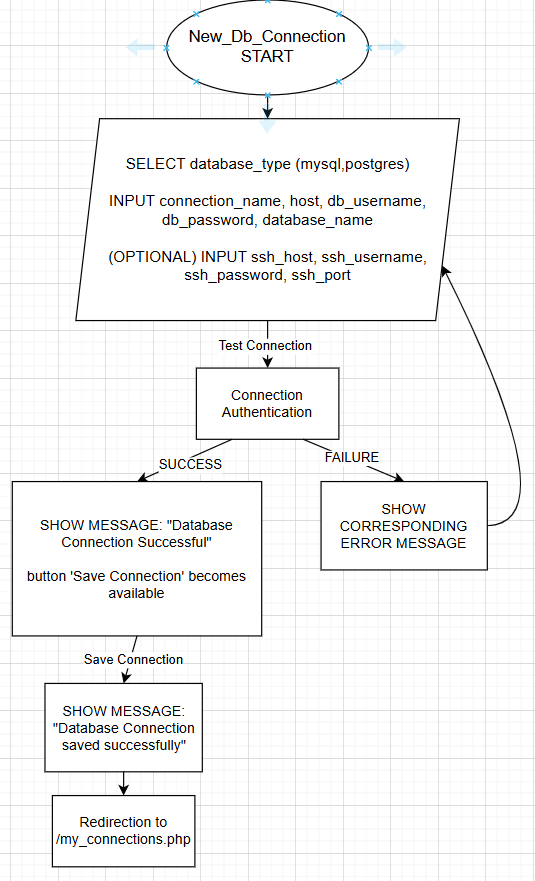
**SIGN-UP FORM**



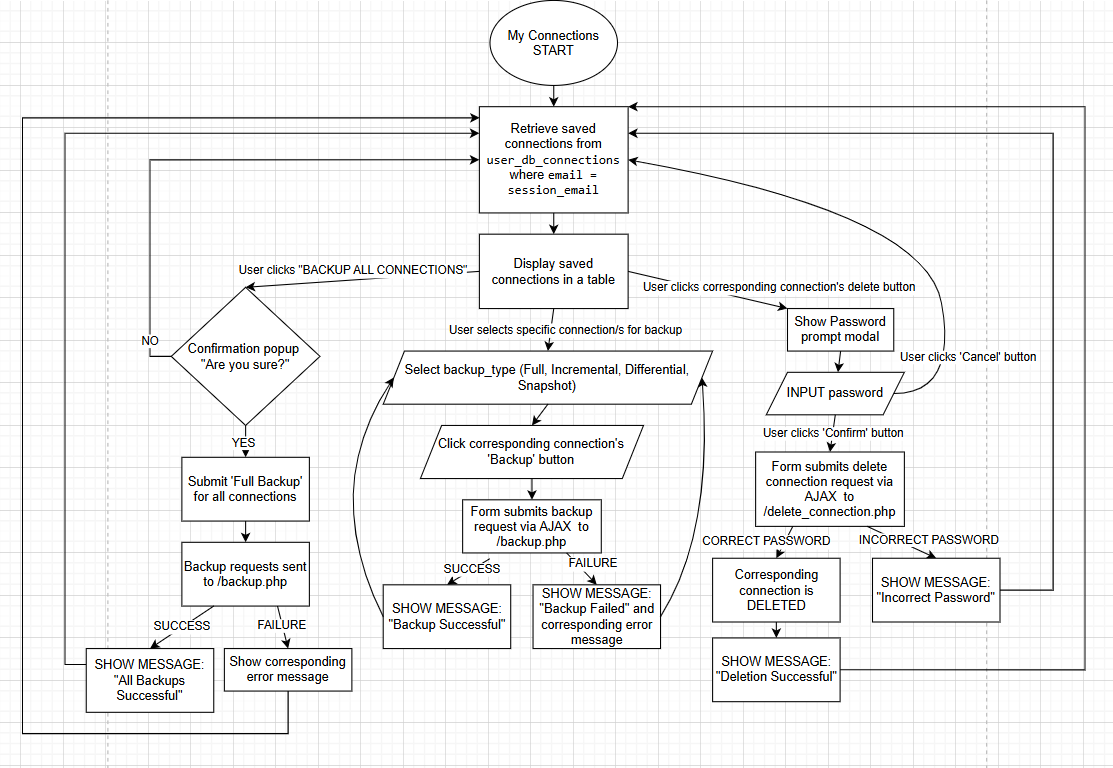
**FORGOT PASSWORD**



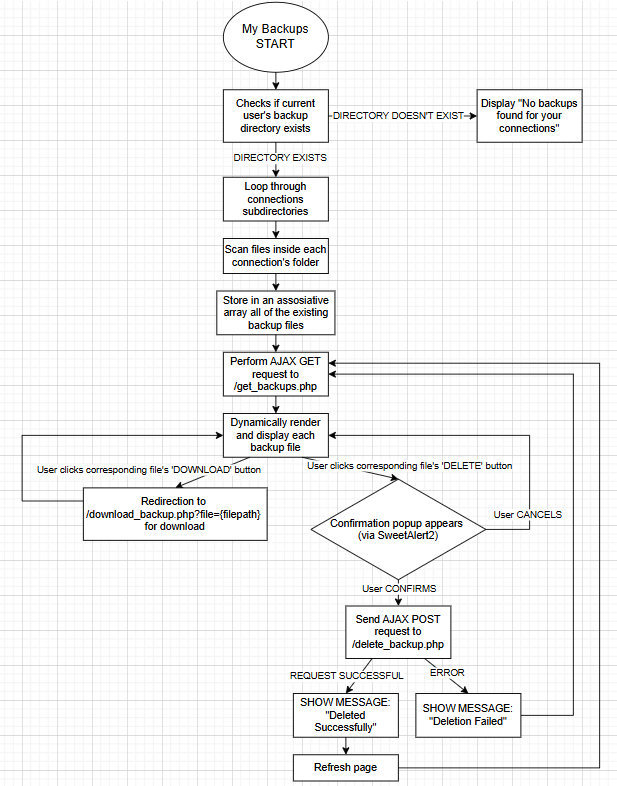
**NEW DATABASE CONNECTION FORM**

****

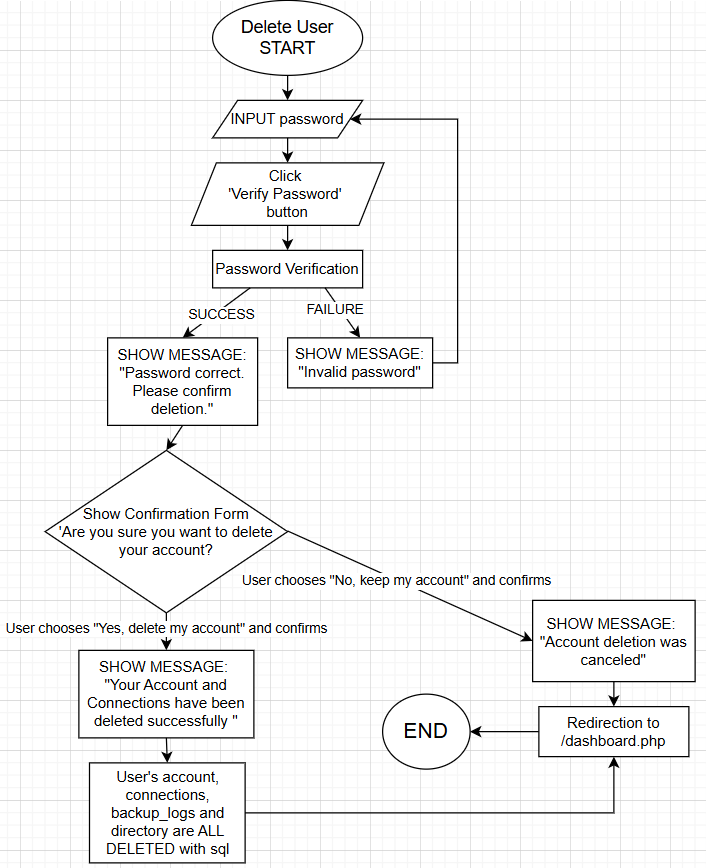
**MY CONNECTIONS**

****

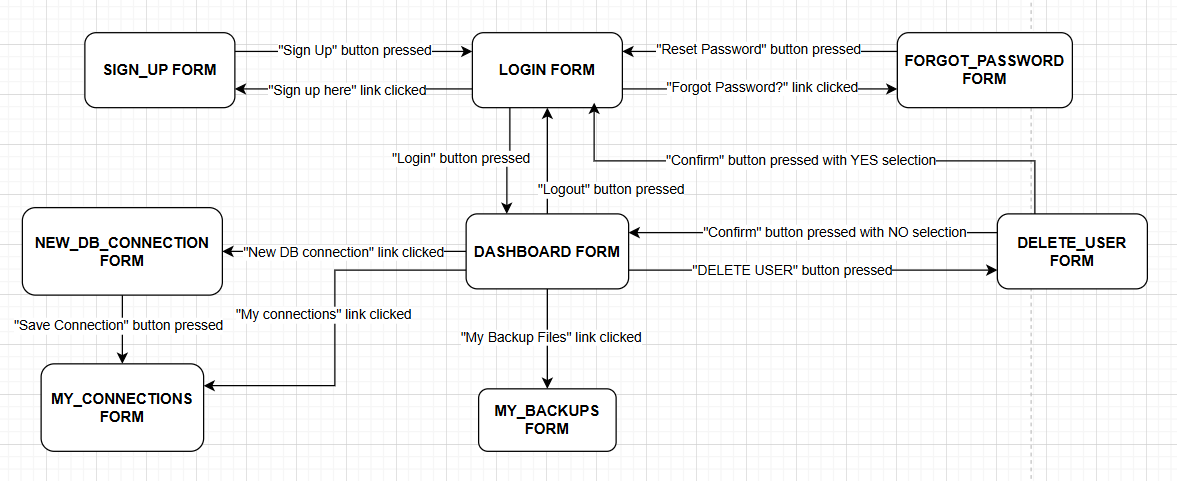
**MY BACKUPS**

****

**DELETE USER FORM**

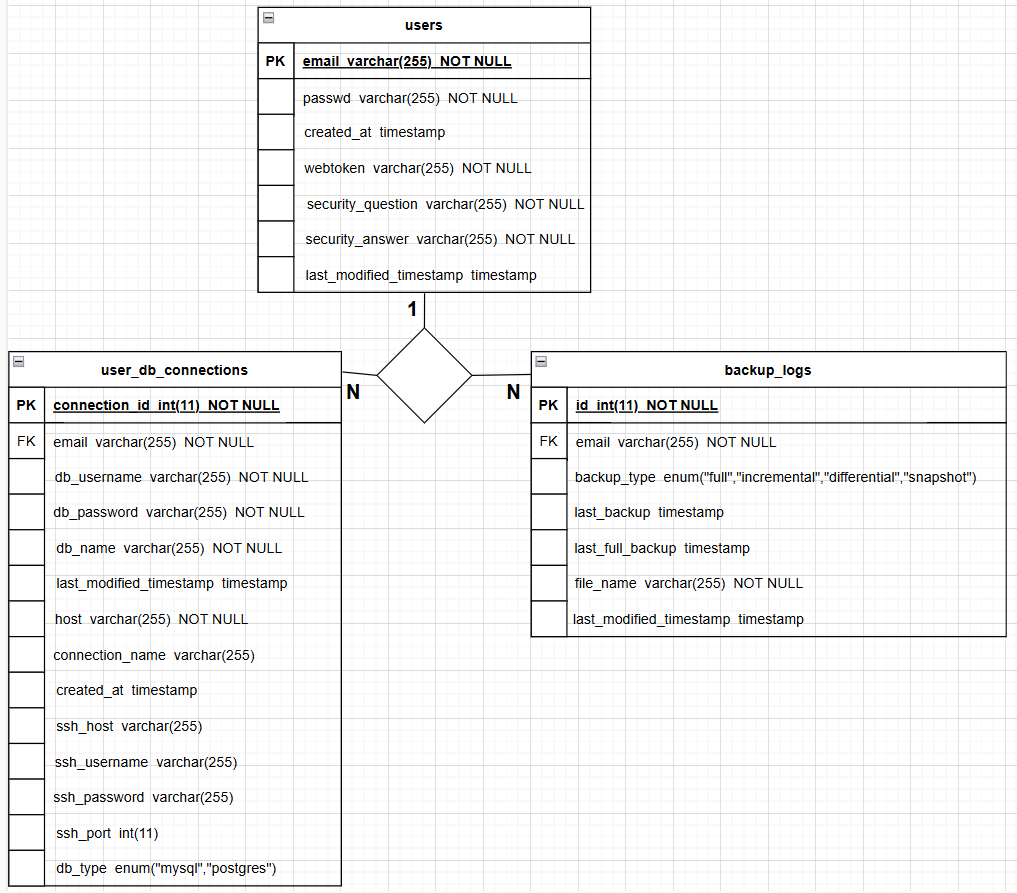
****

**ΓΕΝΙΚΟ ΔΙΑΓΡΑΜΜΑ ΡΟΗΣ ΕΦΑΡΜΟΓΗΣ**



## ER Diagram (Διάγραμμα Σχέσης Οντοτήτων)

Η κύρια λειτουργία της πτυχιακής μου εφαρμογής είναι η δημιουργία και η διαχείριση αντιγράφων ασφαλείας βάσεων δεδομένων. Η εφαρμογή αυτή υποστηρίζει πολλαπλούς παρόχους, χρησιμοποιεί SSH Tunneling για ασφαλείς απομακρυσμένες συνδέσεις και αποθηκεύει τις ρυθμίσεις κάθε χρήστη. Παρακάτω παρουσιάζεται το ER διάγραμμα της βάσης δεδομένων που χρησιμοποίησα για την λειτουργικότητα της εφαρμογής.



Η εφαρμογή μου χρησιμοποιεί 3 πίνακες. Ο πίνακας **users**, που αποθηκεύει τα στοιχεία αυθεντικοποίησης των εκάστοτε χρηστών, έχει σχέση 1 προς Ν με τους άλλους 2 πίνακες, εκ των οποίων ο πίνακας **user\_db\_connections** αποθηκεύει πληροφορίες και στοιχεία αυθεντικοποίησης για τις εκάστοτε συνδέσεις βάσεων δεδομένων του κάθε χρήστη και ο πίνακας **backup\_logs** περιέχει καταγραφές δημιουργίας αντιγράφων ασφαλείας ως ιστορικό, καθώς και πληροφορίες για το είδος των εκάστοτε αντιγράφων που δημιουργήθηκαν.

## Υλοποίηση του FRONT END

Η κυριότερη απαίτηση των χρηστών είναι η ομαλή αλληλεπίδραση του με την εφαρμογή και η εξασφάλιση ότι ανα πάσα χρονική στιγμή θα μπορούν να δημιουργούν ψηφιακά αντίγραφα ασφαλείας. Λαμβάνοντας υπόψη αυτό, πρωταρχικός μου στόχος ήταν να δημιουργήσω ένα περιβάλλον φιλικό προς τον χρήστη, εύκολα κατανοητό, όσο το δυνατόν πιο ευχάριστο αισθητικά και με εξασφαλισμένη την λειτουργικότητα του. Για να επιτευχθεί αυτό, χρησιμοποίησα διάφορες web τεχνολογίες και εργαλεία.

Η βασική δομή των σελίδων δημιουργήθηκε με **HTML5**, ενώ στο κομμάτι της αισθητικής και της διαμόρφωσης της διεπαφής των χρηστών χρησιμοποιήθηκε η **CSS3**. Ιδιαίτερη προσοχή έδωσα στο responsive design και στην σωστή οργάνωση των στοιχείων που εμφανίζονται στις σελίδες, προκειμένου να διευκολύνεται όσο το δυνατόν περισσότερο η περιήγηση του χρήστη στην εφαρμογή.

Στο κομμάτι της αλληλεπίδρασης του με την εφαρμογή, χρησιμοποίησα **JAVASCRIPT**. Με την χρήση αυτού κατάφερα να υλοποιήσω δυναμικές λειτουργίες, όπως την εμφάνιση/απόκρυψη στοιχείων, την επιβεβαίωση πριν την διαγραφή ενός αντιγράφου ή χρήστη αντίστοιχα, καθώς και οποιαδήποτε ενέργεια πραγματοποιείται, χωρίς να απαιτούνται περιττές επαναφορτώσεις των σελίδων.

Ένα ακόμη βασικό κομμάτι του frontend ήταν η ενσωμάτωση της τεχνολογίας **AJAX**, η οποία δίνει την δυνατότητα αποστολής αιτημάτων προς το backend με ασύγχρονο τρόπο. Αυτό σημαίνει ότι ο χρήστης μπορούσε να δημιουργήσει αντίγραφα ασφαλείας, να δει τις αποθηκευμένες του συνδέσεις ή να διαγράψει αρχεία χωρίς να “παγώνει” η σελίδα.

Ο συνδυασμός όλων των παραπάνω τεχνολογιών οδήγησε στην επίτευξη μιας ομαλής και ευχάριστης εμπειρίας κατά την χρήση της εφαρμογής, με όλες τις λειτουργίες της να πραγματοποιούνται επιτυχώς και χωρίς καμία περιττή καθυστέρηση. Το frontend κομμάτι χτίστηκε έτσι ώστε η εφαρμογή να φαίνεται επαγγελματική, συνδυάζοντας ομαλή περιήγηση και υψηλή λειτουργικότητα, με κυριότερα θεμέλια της τον συνδυασμό των απαιτήσεων και των αναγκών των χρηστών.

## Υλοποίηση του BACK END

Το backend κομμάτι της εφαρμογής αποτελεί τον κεντρικό άξονα της δομής της εφαρμογής, καθώς σε αυτό υλοποίησα όλες τις λειτουργίες του χρήστη, τις συνδέσεις με τις βάσεις δεδομένων καθώς και την εκτέλεση των εντολών δημιουργίας αντιγράφων ασφαλείας. Η κύρια γλώσσα που χρησιμοποίησα για την ανάπτυξη του backend ήταν η **PHP**, λόγω της σταθερότητας, της ευρείας υποστήριξης αλλα και της δυνατότητας της να ενσωματώνεται εύκολα με HTML και MySQL.

Χρησιμοποίησα την PHP για την υλοποίηση των λειτουργιών της εγγραφής, της σύνδεσης και της επαναφοράς του κωδικού χρήστη, της αποθήκευσης των στοιχείων των συνδέσεων αλλά και για την εκτέλεση των εντολών δημιουργίας αντιγράφων ασφαλείας βάσεων δεδομένων. Για να δημιουργηθούν τα απαραίτητα αρχεία backup τόσο για MySQL όσο και για PostgreSQL, χρησιμοποίησα τα εργαλεία mysqldump και pg\_dump για κάθε βάση αντίστοιχα, τα οποία εκτελούνται μέσω της PHP.

Για την οργάνωση των φακέλων του κάθε χρήστη, χρησιμοποίησα την εντολή **mkdir**, προκειμένου να δημιουργείται μια δομή αρχείων βασισμένη όπως προανέφερα στο email του κάθε χρήστη και στο όνομα της εκάστοτε σύνδεσης του. Έτσι, κάθε αρχείο backup αποθηκεύεται ταξινομημένα, επιτυγχάνοντας έτσι πιο ξεκάθαρη και αποτελεσματική διαχείριση τους.

Η βάση δεδομένων της ίδιας της εφαρμογής υλοποιήθηκε με την MySQL, με τα εργαλεία της οποίας οργανώθηκαν οι πίνακες των χρηστών, των συνδέσεων και των ιστορικών των αντιγράφων. Ο λόγος που επέλεξα την MySQL έναντι της PostgreSQL στηρίζεται στην μεγαλύτερη ευκολία στην ρύθμιση της και κυρίως στην μεγάλη υποστήριξη της από την PHP. Είναι πιο διαδεδομένη σε κοινά web hosting περιβάλλοντα, γεγονός που με διευκόλυνε ιδιαίτερα στο στήσιμο της και στην ικανοποίηση των λειτουργικών απαιτήσεων της εφαρμογής μου.

Η ασφάλεια των δεδομένων, ύψιστης σημασίας παράγοντας, διασφαλίστηκε μέσω την χρήσης συναρτήσεων password\_hash και το password\_verify, ώστε οι κωδικοί να αποθηκεύονται με ασφάλεια.

Τέλος, μέσω των API endpoints, δόθηκε η δυνατότητα στην εφαρμογή να χειρίζεται με ξεκάθαρο τρόπο τις διάφορες ενέργειες χρηστών που καλούνται από το frontend κομμάτι. Με τον συνδυασμό όλων των παραπάνω εργαλείων και τεχνολογιών, το backend κομμάτι εξασφάλισε την λειτουργικότητα και την αξιοπιστία της εφαρμογής, με επίκεντρο πάντα την ασφάλεια των δεδομένων και την αυξημένη αποδοτικότητα.

## Github Repository

Όλος ο πηγαίος κώδικας της εφαρμογής μου έχει αναρτηθεί σε δημόσιο αποθετήριο στο github.

Μέσα σε αυτό περιέχονται όλα τα αρχεία της εφαρμογής, χωρισμένα σε φακέλους ανάλογα με την λειτουργία τους (frontend, backend, logs, docs), για να διευκολύνεται η κατανόηση της δομής του προγράμματος.

Η ύπαρξη του αποθετηρίου επιτρέπει την επαναχρησιμοποίηση ή επέκταση της εφαρμογής από άλλους χρήστες στο μέλλον , ενώ παράλληλα λειτουργούσε ως τεκμήριο της προόδου και της μεθοδολογίας και του συνόλου των διαφορετικών τεχνολογιών που χρησιμοποίησα.

Τό αποθετήριο της εφαρμογής μου είναι διαθέσιμο στον παρακάτω σύνδεσμο:

# Συμπεράσματα ή/και προτάσεις βελτίωσης

Σε αυτήν την ενότητα αναλύονται τα συμπεράσματα της διπλωματικής εργασίας και παρουσιάζονται προτάσεις βελτίωσης της.

# ΒΙΒΛΙΟΓΡΑΦΙΑ

Ο τρόπος γραφής των βιβλιογραφικών αναφορών γίνεται σύμφωνα με τα παρακάτω παραδείγματα (IEEE style):

**Βιβλία**

[] W. K. Chen, *Linear Networks and Systems*. Belmont, CA: WadsworthPress, 2003.

[] J. L. Spudich and B. H. Satir, *Sensory Receptors and Signal Transduction*. New York: Wiley-Liss, 2001.

**Application Note**

[] Hewlett-Packard, Appl. Note 935, pp.25-29.

**Πατέντες**

[] K. Kimura and A. Lipeles, “Fuzzy controller component,” U. S. Patent 14, 860,040, 14 Dec., 2006.

**Data Sheet**

[] Texas Instruments, “High speed CMOS logic analog multiplexers/demultiplexers,” 74HC4051 datasheet, Nov. 1997.

**Internet Site**

[4] <https://dev.mysql.com/doc/refman/8.4/en/mysqldump.html>

[5] <https://www.postgresql.org/docs/8.1/app-pgdump.html>

[6] <https://www.dbvis.com/thetable/a-complete-guide-to-pg-dump-with-examples-tips-and-tricks/>

[7]<https://docs.aws.amazon.com/prescriptive-guidance/latest/migration-large-mysql-mariadb-databases/mysqldump-and-mysqlpump.html>

[8]<https://docs.redhat.com/fr/documentation/red_hat_enterprise_linux/9/html/configuring_and_using_database_servers/backing-up-postgresql-data_using-postgresql#advantages_and_limitations_of_file_system_backing_up>

[9]<https://www.depesz.com/2019/12/10/how-to-effectively-dump-postgresql-databases/>

[10]<https://dev.mysql.com/doc/refman/8.4/en/backup-types.html>

[12]<https://dev.mysql.com/doc/mysql-backup-excerpt/8.0/en/mysqldump-sql-format.html>

[13]<https://www.postgresql.org/docs/current/backup-dump.html>

[14]<https://scalegrid.io/blog/mysql-backups-best-practices/>

[15]<https://www.ibm.com/docs/en/netezza?topic=nc-incremental-backups-2>

[16]<https://aatalrashid.medium.com/mysql-incremental-backup-using-binary-logs-bin-logging-42473adac650>

[17]<https://pgbackrest.org/user-guide.html#concept/backup>

[18]<https://www.postgresql.org/docs/current/continuous-archiving.html>

[19]<https://www.ubackup.com/enterprise-backup/mysql-differential-backup-0020.html>

[20]<https://www.postgresql.org/docs/current/app-pgbasebackup.html>

[21]<https://www.rubrik.com/insights/what-is-a-snapshot-backup>

[22]<https://www.lullabot.com/articles/mysql-backups-using-lvm-snapshots>

[23]<https://dbsnapper.com/blog/how-to-create-a-postgresql-database-snapshot>

[24]<https://developer.mozilla.org/en-US/docs/Web/HTML>

[29][https://www.w3docs.com/learn-html/html-styles.html](https://www.w3docs.com/learn-html/html-styles.html?utm_source=chatgpt.com)

[32]<https://www.w3schools.com/php/php_intro.asp>

[35]<https://www.w3schools.com/php/php_error.asp>

[38]<https://learntocodewith.me/posts/sql-guide/>

[39]<https://www.astera.com/knowledge-center/sql-vs-nosql/>

**Paper in Conference Proceedings**

[] J. Smith, R. Jones, and K. Trello, “Adaptive filtering in data communications with self-improved error reference,” In Proc. IEEE International Conference on Wireless Communications ’04, 2004, pp. 65-68.

[] H. A. Nimr, “Defuzzification of the outputs of fuzzy controllers,” presented at 5th International Conference on Fuzzy Systems, Cairo, Egypt, 2006.

**Journal Articles**

[1]<https://www.ekriti.gr/business/psifiaki-diaheirisi-eggrafon-giati-prepei-na-arnitheite-ta-entypa-eggrafa>

[2]<https://www.zisopoulos.com/blog/124-psifiakos-metaskhimatismos-pleonektimata-kai-meionektimata>

[3]<https://www.zisopoulos.com/blog/60-prokhorwntas-se-ena-paperless-khoris-kharti-ghrafio>

[11]<https://medium.com/@masoud.chelongar/postgresql-backup-types-comparison-contrast-b4945177cabc>

[25]<https://unstop.com/blog/advantages-and-disadvantages-of-html>

[26]<https://www.getresponse.com/blog/tips-and-tools-to-write-seo-friendly-content>

[27]<https://www.geeksforgeeks.org/advantages-and-disadvanatges-of-html/>

[28]<https://developer.mozilla.org/en-US/docs/Web/API/Service_Worker_API/Using_Service_Workers>

[30][https://primercss.io/what-is-external-css/](https://primercss.io/what-is-external-css/?utm_source=chatgpt.com)

[31]<https://www.geeksforgeeks.org/advantages-and-disadvantages-of-css/>

[33]<https://www.geeksforgeeks.org/advantages-and-disadvantages-of-php/>

[34]<https://customlab.gr/php-advantages-disadvantages/>

[37]<https://projectmanagers.net/top-10-pros-and-cons-of-using-php/>

[40]<https://www.geeksforgeeks.org/what-are-the-advantages-and-disadvantages-of-using-sql-vs-nosql-databases/>

1. **: ΤΙΤΛΟΣ ΠΑΡΑΡΤΗΜΑΤΟΣ**

Τα παραρτήματα μπορούν να είναι περισσότερα από ένα. Αριθμούνται με γράμματα του Ελληνικού αλφάβητου (Α, Β, Γ, …).

Στα παραρτήματα παρουσιάζονται πληροφορίες που δεν είναι κρίσιμες για την εργασία, αλλά σημαντικές για την απόδειξη συμπερασμάτων που αναπτύχθηκαν στην εργασία. Περιέχουν κώδικα λογισμικού, ερωτηματολόγια και απαντήσεις σε ερωτηματολόγια, κτλ.