

Δημιουργία Βάσης Δεδομένων για Κλινική με Επιλογή Ραντεβού

Ονόματα Μελών Ομάδας:

Άγγελος Κονταλής

Νικολάος Χάχαλης

Εισαγωγή

Εισαγωγή

Στη σύγχρονη εποχή, η οργάνωση και η διαχείριση των ραντεβού σε μία κλινική αποτελεί μία από τις πιο απαιτητικές διαδικασίες. Η ανάγκη για ακρίβεια, αποφυγή διπλοκρατήσεων και αποτελεσματική επικοινωνία μεταξύ των ασθενών και του προσωπικού καθιστούν απαραίτητη την υιοθέτηση λύσεων που αξιοποιούν την τεχνολογία. Με βάση αυτό το σκεπτικό, η παρούσα εργασία επικεντρώνεται στη δημιουργία ενός προγράμματος που λειτουργεί παρόμοια με το Doodle, διευκολύνοντας τη διαχείριση των ραντεβού μίας κλινικής. Ο στόχος είναι η αυτοματοποίηση της διαδικασίας κράτησης ραντεβού, η οποία θα επιτρέπει στους χρήστες να καταχωρούν τα προσωπικά τους δεδομένα και να επιλέγουν μια διαθέσιμη ώρα για το ραντεβού τους.

Η διαδικασία υλοποίησης της εφαρμογής βασίστηκε σε συνεργασία και προσεκτικό σχεδιασμό. Αρχικά, συζητήσαμε και καταλήξαμε στη δημιουργία ενός **Entity-Relationship Diagram (ERD)**, το οποίο αποτέλεσε τη βάση για την ανάπτυξη του σχήματος της βάσης δεδομένων. Στη συνέχεια, το σχήμα αυτό μεταφέρθηκε σε περιβάλλον **MySQL**, όπου δημιουργήθηκε η ίδια η βάση δεδομένων. Κατόπιν, χρησιμοποιήθηκε η γλώσσα προγραμματισμού **Python**, σε συνδυασμό με τη βιβλιοθήκη **PyMySQL**, για την ανάπτυξη του προγράμματος, το οποίο περιλαμβάνει διεπαφή χρήστη (interface) που καθιστά τη χρήση του εύκολη και λειτουργική.

Η εργασία μας είναι σημαντική διότι παρέχει μία ολοκληρωμένη και ευέλικτη λύση για τη διαχείριση ραντεβού, βελτιώνοντας την αποδοτικότητα των λειτουργιών μίας κλινικής. Το πρόγραμμα προσφέρει πληθώρα λειτουργιών, όπως τη δημιουργία νέων ραντεβού, την τροποποίηση ή ακύρωσή τους, την εμφάνιση διαθέσιμων χρονικών διαστημάτων, καθώς και την εύκολη αναζήτηση προηγούμενων καταχωρήσεων. Μέσω αυτών των χαρακτηριστικών, όχι μόνο μειώνεται ο φόρτος εργασίας του προσωπικού, αλλά και διασφαλίζεται η καλύτερη εμπειρία των ασθενών.

Η βασική ιδέα του πρότζεκτ είναι η βελτίωση της αποδοτικότητας και της ευχρηστίας του συστήματος, καθώς και η μείωση των λαθών στην καταχώρηση των δεδομένων.

Η ανάπτυξη του συγκεκριμένου έργου αποτελεί μία προσέγγιση που συνδυάζει την τεχνική κατάρτιση με τη δημιουργικότητα, ενώ αναδεικνύει τη δύναμη της συνεργασίας στην παραγωγή ποιοτικών και λειτουργικών λύσεων.

Μεθοδολογία

Η ανάπτυξη του προγράμματος διαχείρισης ραντεβού της κλινικής βασίστηκε σε μια συστηματική μεθοδολογία, η οποία περιλάμβανε τα παρακάτω βήματα:

1. Συζήτηση του μικρόκοσμου και των χαρακτηριστικών του

Το πρώτο στάδιο της εργασίας επικεντρώθηκε στην κατανόηση του μικρόκοσμου της κλινικής και των απαιτήσεών του. Αναζητήσαμε πληροφορίες στο διαδίκτυο ενώ παράλληλα πραγματοποιήσαμε συζητήσεις με άτομα που δουλεύουν σε κλινικές για να καταγράψουμε τα βασικά χαρακτηριστικά και τις λειτουργίες που απαιτούνται, όπως:

- Η δυνατότητα καταχώρησης και διαχείρισης ραντεβού.
- Ο ορισμός χρονικών διαστημάτων διαθεσιμότητας.
- Η αποφυγή διπλοκρατήσεων στα ραντεβού.
- Η δυνατότητα τροποποίησης ή ακύρωσης ραντεβού.
- Η αναζήτηση προηγούμενων ραντεβού με βάση διάφορα κριτήρια.
- Η απλότητα και ευχρηστία της διεπαφής για το προσωπικό και τους χρήστες.

Με βάση αυτά τα χαρακτηριστικά, καθορίσαμε το πλαίσιο λειτουργίας του προγράμματος και θέσαμε τους στόχους υλοποίησης.

2. Σχεδιασμός του ERD (Entity-Relationship Diagram)

Στο επόμενο στάδιο, σχεδιάσαμε το **ERD**, το οποίο αποτελεί τη γραφική αναπαράσταση των οντοτήτων του μικρόκοσμου και των σχέσεων μεταξύ τους. Το ERD περιελάμβανε:

- Την οντότητα *Ασθενής* (Patient), με πεδία Όνομα, επώνυμο, τηλέφωνο, email, ΑΜΚΑ, διεύθυνση και ημερομηνία γέννησης.
- Την οντότητα *Ραντεβού* (Appointment), με πεδία όπως η ημερομηνία, η ώρα, ο γιατρός, ο ασθενής και το κλειδί του σλοτ.

- Την οντότητα *Γιατρός* (Physician), με πληροφορίες για το όνομα, την ειδικότητα και άλλα στοιχεία του γιατρού.
- Την οντότητα (Timeslot) με πληροφορίες για το νούμερο, την ημερομηνία, το κλειδί του slot και το AMKA του ασθενή.
- Τις σχέσεις μεταξύ αυτών των οντοτήτων, π.χ., κάθε ραντεβού συνδέεται με έναν ασθενή και έναν γιατρό.

3. Δημιουργία του Σχήματος (Schema)

Αφού ολοκληρώσαμε το ERD, προχωρήσαμε στον σχεδιασμό του **σχήματος της βάσης δεδομένων**. Δημιουργήσαμε πίνακες για κάθε οντότητα και καθορίσαμε τα πεδία, τους τύπους δεδομένων και τις σχέσεις μεταξύ των πινάκων, όπως:

- Τα πρωτεύοντα και ξένα κλειδιά.
- Οι περιορισμοί ακεραιότητας δεδομένων.
- Οι σύνδεσμοι μεταξύ των πινάκων, όπως και οι συσχετίσεις τους.

4. Υλοποίηση της Βάσης Δεδομένων στη MySQL

Με το σχήμα έτοιμο, προχωρήσαμε στη δημιουργία της βάσης δεδομένων χρησιμοποιώντας τη **MySQL**. Πραγματοποιήσαμε τα παρακάτω βήματα:

- Δημιουργία της βάσης δεδομένων με τις εντολές SQL.
- Δημιουργία των πινάκων σύμφωνα με το σχήμα που σχεδιάσαμε.
- Εισαγωγή δείγματος δεδομένων για σκοπούς δοκιμών.
- Εξασφάλιση της σωστής λειτουργίας των συνδέσεων και των περιορισμών.

5. Ανάπτυξη του Backend και της Διεπαφής Χρήστη με Python

Το τελικό στάδιο περιλάμβανε την ανάπτυξη του κώδικα στην **Python**, με χρήση της βιβλιοθήκης **PyMySQL** για τη σύνδεση και την αλληλεπίδραση με τη βάση δεδομένων. Οι κύριες ενέργειες που πραγματοποιήθηκαν ήταν:

- Δημιουργία των συναρτήσεων για την καταχώρηση, τροποποίηση, ακύρωση, αναζήτηση ραντεβού αλλά και πολλών άλλων.
- Υλοποίηση μηχανισμού για την αποφυγή διπλοκρατήσεων.

- Ανάπτυξη της διεπαφής χρήστη, που επιτρέπει στο προσωπικό να αλληλεπιδρά εύκολα με το σύστημα.
- Δοκιμή της εφαρμογής με πραγματικά δεδομένα για να διασφαλιστεί η σωστή λειτουργία όλων των χαρακτηριστικών.

Αξιολόγηση

Για την αξιολόγηση του συστήματος, χρησιμοποιήθηκαν τα εξής κριτήρια επιτυχίας:

1. Ελέγξαμε αν τα δεδομένα καταχωρούνται σωστά στη βάση και αν όλες οι λειτουργίες του προγράμματος δουλεύουν όπως είχαμε σχεδιάσει.
2. Η ταχύτητα απόκρισης του συστήματος και η αποτελεσματικότητα στην αναζήτηση διαθέσιμων ραντεβού.
3. Η διεπαφή χρήστη αξιολογήθηκε ως προς την ευχρηστία της.
4. Κρίναμε απαραίτητη τη χρήση ρεαλιστικών δεδομένων.
5. Σημαντικό για την ομάδα μας φάνηκε η υλοποίηση πιο δημιουργικών συναρτήσεων.

Δεδομένα

Η βάση δεδομένων περιλαμβάνει τα εξής δεδομένα με προσοχή στο να είναι ρεαλιστικά:

- Προσωπικά δεδομένα χρηστών: Όνομα, επώνυμο, τηλέφωνο, email, AMKA, διεύθυνση και ημερομηνία γέννησης.
- Διαθεσιμότητες γιατρών: Οι ώρες που είναι διαθέσιμοι για ραντεβού, τα κλειδιά των γιατρών και τα κλειδιά των slots.
- Ραντεβού: το κλειδί του ραντεβού, η ημερομηνία, η κατάσταση, το AMKA του ασθενή και το κλειδί του γιατρού.
- Κριτικές ασθενών : Η βαθμολογία, το AMKA του ασθενή, το κλειδί του γιατρού και το ραντεβού για το οποίο βαθμολογείται.
- Γιατροί: Το κλειδί του εκάστοτε γιατρού, το email του, το επίθετο, το όνομα, το τηλέφωνο και η ειδικότητα του.
- Τα Timeslots: το νούμερο, η ημερομηνία, το κλειδί του slot και το AMKA του ασθενή.

Κύριες Ενέργειες και Συνεργασία Ομάδας

Συνεργασία Ομάδας

Η εργασία αυτή βασίστηκε σε ένα μοντέλο συνεργασίας που χαρακτηρίστηκε από συνεχή επικοινωνία, ανταλλαγή ιδεών και παράλληλη εργασία. Η ομάδα μας λειτούργησε αποτελεσματικά μέσα από τακτικές κλήσεις, όπου συζητούσαμε, ανταλλάσσαμε απόψεις και λαμβάναμε αποφάσεις από κοινού.

Ένας από τους πιο σημαντικούς παράγοντες της επιτυχίας μας ήταν η δυνατότητα να εργαζόμαστε ταυτόχρονα και συντονισμένα. Κάθε βήμα της εργασίας συνοδευόταν από εκτενείς συζητήσεις, όπου αναλύαμε τις απαιτήσεις, προτείναμε λύσεις και συμφωνούσαμε για τον τρόπο υλοποίησης.

Σχεδιασμός ERD και Schema

Το πρώτο τεχνικό βήμα ήταν η δημιουργία του **Entity-Relationship Diagram (ERD)**, το οποίο αποτέλεσε το θεμέλιο για την κατασκευή της βάσης δεδομένων. Όπου κατασκευάσαμε αρκετές εκδόσεις μέχρι να συμφωνήσουμε και να καταλήξουμε στο τελικό μας μοντέλο.

Αφού ολοκληρώσαμε το ERD, μετατρέψαμε το διάγραμμα σε **σχήμα βάσης δεδομένων (schema)**. Οπου προσθέσαμε τα απαραίτητα κλειδιά, τις συσχετίσεις αλλά και τους περιορισμούς.

Υλοποίηση στη MySQL

Ακολουθώντας το schema, προχωρήσαμε στη δημιουργία της βάσης δεδομένων στη MySQL. Η υλοποίηση της βάσης δεδομένων έγινε επίσης μέσα από συνεχείς συζητήσεις. Εργαστήκαμε συνεργατικά.

Εργασία στον Κώδικα της Python

Η ανάπτυξη του κώδικα στην Python ήταν μια διαδικασία που πραγματοποιήθηκε σχεδόν εξολοκλήρου μέσα από συνεχή διάλογο. Κατά τη διάρκεια των κλήσεων εντοπίζαμε τις βασικές συναρτήσεις που χρειάζεται να υλοποιηθούν, όπως η καταχώρηση, η τροποποίηση και η ακύρωση ραντεβού, αλλά και οι έλεγχοι για διαθεσιμότητα. Συζητούσαμε τη δομή κάθε συνάρτησης, τα δεδομένα εισόδου, την επεξεργασία και την έξοδο. Κάθε συνάρτηση σχεδιάστηκε με προσοχή στις λεπτομέρειες, με στόχο τη σωστή επικοινωνία με τη βάση δεδομένων μέσω της PyMySQL. Αναλάμβανε ο καθένας να γράψει εναλλάξ συγκεκριμένες συναρτήσεις, με τον άλλο να παρακολουθεί και να παρέχει σχόλια ή ιδέες σε πραγματικό χρόνο. Αυτή η διαδικασία βοήθησε στη διασφάλιση της συνοχής του κώδικα και στη συνεχή βελτίωσή του.

Ανταλλαγή Ιδεών

Η συνεχής ανταλλαγή ιδεών ήταν κεντρικό στοιχείο της συνεργασίας μας. Κάθε φορά που συναντούσαμε μια πρόκληση, είτε στο σχεδιασμό του backend είτε στη λειτουργικότητα της διεπαφής χρήστη, συζητούσαμε εκτενώς τις πιθανές λύσεις. Οι διαφορετικές προσεγγίσεις μας όχι μόνο έλυναν προβλήματα, αλλά συχνά οδηγούσαν σε βελτιώσεις που δεν είχαμε αρχικά σκεφτεί.

Χρονοδιάγραμμα

- 25/10/2024 - 10/11/2024: Αναλογισμός μικρόκοσμου και σχεδίαση ERD μοντέλου.
- 11/11/2024 - 30/11/2024: Ανάπτυξη schema και βάση δεδομένων.
- 10/12/2024 - 03/01/2025: Ανάπτυξη backend αλλά και διεπαφής χρήστη, σύνδεση με τη βάση δεδομένων.
- 03/01/2024 - 05/01/2025: Δοκιμές απόδοσης και τελική αξιολόγηση του συστήματος.
- 06/01/2025: Ολοκλήρωση αναφοράς και παραδοτέων.

Συμπεράσματα

Το πρότζεκτ ολοκληρώθηκε με επιτυχία, καταφέροντας να δημιουργήσει μια λειτουργική βάση δεδομένων και εφαρμογή για την κράτηση ραντεβού σε κλινική. Η εφαρμογή είναι εύκολη στη χρήση και ανταποκρίνεται στις ανάγκες της κλινικής για διαχείριση των ραντεβού. Στο μέλλον, η εφαρμογή μπορεί να επεκταθεί με περισσότερες δυνατότητες, όπως υπενθυμίσεις για ραντεβού και προσθήκη συστημάτων πληρωμής.

Παραρτήματα

1. Οδηγίες Εγκατάστασης:

- Κατεβάστε τα απαραίτητα αρχεία (clinic.py, clinic_db.sql).
- Εγκαταστήστε τις βιβλιοθήκες με την εντολή:
``pip install -r PyMySQL`, `pip install -r tabulate``
- Πρέπει να γίνει host η βάση δεδομένων.

Πχ. Με χρήση του xampp, με local host.

- Εκτελέστε το πρόγραμμα και περιηγηθείτε στην διεπαφή.

2. Παραδείγματα Χρήσης:

- Με την εκκίνηση του προγράμματος εμφανίζονται αυτές οι 4 επιλογές.

```
Successfully connected to the database!
Timeslots successfully generated for the upcoming 4 weeks.
(1) Patient
(2) Physician
(3) Data
(0) Exit
```

Εαν:

Επιλέξουμε το (0) τότε το πρόγραμμα τερματίζεται.

Επιλέξουμε το (1):

```
1
Enter your SSN (11-digit integer):
```

Τότε ζητάτε από τον ασθενή να εισάγει το ΑΜΚΑ του με το οποίο ελέγχουμε εάν είναι ήδη εγγεγραμμένος και ανάλογα συνεχίζει το πρόγραμμα. Εάν δεν είναι εγγεγραμμένος τότε του εμφανίζονται τα παρακάτω με τους κατάλληλους ελέγχους ώστε να μην είναι λανθασμένα τα δεδομένα:

```
Do you want to create a new account?
(1) Yes
(0) No
1
Enter your e-mail: papadopoulos@gmail.com
Enter your last name: papadopoulos
Enter your first name: vasilis
Enter your birth date (YYYY-MM-DD): 2002-12-26
Enter your phone number: 6972579296
Enter your address: lefkosias 14
Account created successfully!
```

Εάν ο ασθενής έχει ήδη λογαριασμό τότε του εμφανίζονται οι ακόλουθες επιλογές:

```
Welcome back, Petros!  
Monday, 06-01-2025, 13:15  
(1) Book appointment  
(2) Cancel appointment  
(3) Rate appointment  
(4) Appointment history  
(5) Change data  
(0) Go back
```

Όπου του δίνεται η δυνατότητα να κάνει όλα τα παραπάνω. Πιο αναλυτικά:

- (1) Ζητάει από τον χρήστη ημερομηνία και έπειτα του εμφανίζει όλους τους γιατρούς και τις ειδικότητες τους. Στην συνέχεια αφού επιλέξει τον γιατρό εμφανίζει την διαθεσιμότητα του.
- (2) Εμφανίζει τα τρέχοντα ραντεβού αριθμημένα. Με την επιλογή του αντίστοιχου αριθμού μπορεί να ακυρώσει το ραντεβού του. Προφανώς και η διαθεσιμότητα του γιατρού επανέρχεται για την συγκεκριμένη ημερομηνία και ώρα ώστε να είναι λειτουργική η κλινική.
- (3) Εμφανίζει στον ασθενή όλα τα ολοκληρωμένα του ραντεβού (εάν έχει) και του δίνει την δυνατότητα να τα βαθμολογήσει από το 1-5
- (4) Εμφανίζει στον ασθενή όλο το ιστορικό των ολοκληρωμένων ραντεβού του.
- (5) Με αυτήν την επιλογή ο ασθενής μπορεί να επιλέξει να αλλάξει οποιοδήποτε στοιχείο έχει δηλώσει λανθασμένα.

Επιλέξουμε το (2):

Ζητείται από τον Γιατρό να εισάγει το προσωπικό του κλειδί. Με τον ίδιο τρόπο με πριν ελέγχεται εάν υπάρχει ήδη στην βάση. Εάν δεν υπάρχει, τότε μπορεί να κάνει εγγραφή με τα στοιχεία του. Εάν όμως υπάρχει του εμφανίζει τα ακόλουθα:

```
Welcome back, Nikolas!  
Monday, 06-01-2025, 13:31  
(1) Update availability  
(2) List pending appointments  
(3) Cancel appointment  
(4) Mark appointments  
(5) Create timeslots for specific day  
(6) Hours worked  
(0) Go back
```

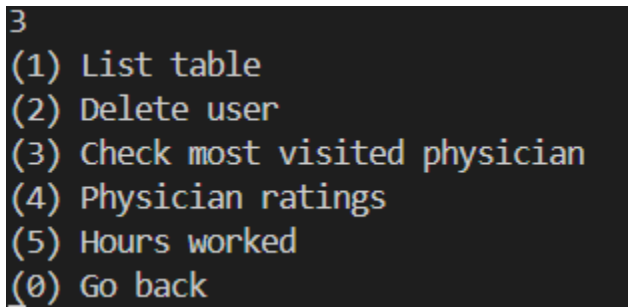
Όπου του δίνεται η δυνατότητα να κάνει όλα τα παραπάνω. Πιο αναλυτικά:

- (1) Εμφανίζονται όλα τα slots που είναι διαθέσιμος. Επιλέγοντας όσα θέλει από αυτά μπορεί να εναλλάξει την διαθεσιμότητα του. Εξ ορισμού όλοι οι

γιατροί είναι διαθέσιμοι στο θωρο που δουλεύει η κλινική εκτός εάν το αλλάξουν οι ίδιοι.

- (2) Εμφανίζονται όλα τα αναμενόμενα ραντεβού του γιατρού.
- (3) Εμφανίζονται όλα τα αναμενόμενα ραντεβού του γιατρού αριθμημένα και μπορεί να ακυρώσει οποιοδήποτε από αυτά.
- (4) Εμφανίζονται όλα τα ραντεβού του γιατρού που είναι πριν την σημερινή ημερομηνία και μπορεί να δηλώσει εάν ολοκληρώθηκαν.
- (5) Εμφανίζει στον γιατρό να εισάγει μια ημερομηνία (εάν δεν πέφτει σε σαββατοκύριακο) δημιουργεί θωρη διαθεσιμότητα για εκείνη την ημερομηνία.
- (6) Εμφανίζει της ώρες που δούλεψε τον μήνα και το έτος που επιλέγει

Επιλέξουμε το (3):



- (1) Αφήνει τον χρήστη να δει όποιον πίνακα της βάσης θέλει.
- (2) Αφήνει τον χρήστη να διαγράψει όποιο χρήστη της βάσης θέλει. Ασθενή ή Γιατρό.
- (3) Τυπώνει τον γιατρό με τα περισσότερα ολοκληρωμένα ραντεβού.
- (4) Εμφανίζει όλους τους γιατρούς και αφού επιλέξει τον γιατρό που θέλει ο χρήστης, του τυπώνει την βαθμολογία του.
- (5) Λειτουργεί με τον ίδιο τρόπο όπως στον γιατρό που αναλύσαμε νωρίτερα. Η χρήση μέσω του data υποδεικνύει ότι χρησιμοποιείται για να δει η γραμματεία ή η διεύθυνση τις αντίστοιχες πληροφορίες.

3. Σύνδεσμος Κώδικα:

[GitHub Repository Link] <https://github.com/AngelosKontalis/PYTHON-SQL-CLINIC-GROUP22>