

# Ομαδική Εργασία στις Βάσεις Δεδομένων

## Εφαρμογή Διαχείρισης Αλυσίδας Πρατηρίων

### Ομάδα 36

Ιωάννης Νικολάου

[up1072681@upnet.gr](mailto:up1072681@upnet.gr) - 1072681

Αλέξανδρος Τσαπάρας

[up1072824@upnet.gr](mailto:up1072824@upnet.gr) - 1072824

#### 1 ΠΕΡΙΛΗΨΗ

Η βασική ιδέα του project ήταν η αποτελεσματική διαχείριση μιας αλυσίδας πρατηρίων στην Ελλάδα ώστε ο ιδιοκτήτης της επιχείρησης να έχει στη διάθεσή του όλες τις απαραίτητες πληροφορίες για αυτήν. Το ενδιαφέρον είναι η ανάλυση της λειτουργίας της αλυσίδας πρατηρίων που ενώ εκ πρώτης όψεως φαίνεται απλή εν τέλει κρύβει αρκετή πολυπλοκότητα. Για την διαχείριση της επιχείρησης του, ο ιδιοκτήτης συνδέεται σε μια σελίδα με γραφική διεπαφή που τον διευκολύνει για την αποτελεσματική διαρρύθμιση της, η οποία γραφική διεπαφή έχει δημιουργηθεί με σύγχρονες τεχνολογίες.

#### 2 ΜΕΘΟΔΟΛΟΓΙΑ

Αρχικά αφού λάβαμε υπόψιν διάφορες πηγές [1],[6] ώστε να κατανοήσουμε καλύτερα τις απαιτήσεις που έχει μια αλυσίδα πρατηρίων, καταγράψαμε τις βασικές οντότητες και τα γνωρίσματά τους. Έπειτα ο καθένας δημιούργησε το δικό του erd μέσω του ErdMaker [4] και μετά από μια μεταξύ μας συνάντηση καταλήξαμε στο αρχικό πρωτότυπο. Στη συνέχεια από την παρεμβολή της ενδιάμεσης παρουσίασης των εργασιών, ανακαλύψαμε μερικές ελλείψεις που είχε η αρχική προσέγγισή μας και τις διορθώσαμε. Το επόμενο βήμα μας ήταν να επισκεφθούμε ένα πρατήριο bp για να συλλέξουμε περισσότερες πληροφορίες, ώστε να συνεχίσουμε στην δημιουργία τυχαίων δεδομένων για την βάση μας μέσω του Mockaroo [3]. Έπειτα ελέγξαμε τα δεδομένα, ώστε να συμφωνούν μεταξύ τους και να πληρούν τους περιορισμούς που έχουμε θέσει στο Σχεσιακό Μοντέλο, που δημιουργήσαμε μέσω του DB Designer [5] και που γενικά ισχύουν σε μια αλυσίδα πρατηρίων. Το τελευταίο βήμα μας ήταν να δημιουργήσουμε το repository της εργασίας μέσω του GitHub [2], το οποίο θα αποτελείται από τις βασικές ενέργειες του κάθε table - CRUD (create, read, update, delete), περιορισμούς και τα απαραίτητα αρχεία για την γραφική διεπαφή της και ακόμα ένα Balance Sheet, το οποίο θα διευκολύνει τον ιδιοκτήτη να κατανοήσει καλύτερα την επίδοση της επιχείρησης του και πιο συγκεκριμένα του κάθε πρατηρίου.

## 2.1 Το ενδιαφέρον του project

Το ενδιαφέρον του project είναι τα περίπλοκα ερωτήματα που μπορεί να εκτελέσει ένας χρήστης αυτής της εφαρμογής (π.χ. ιδιοκτήτης) και να λάβει χρήσιμες πληροφορίες για τη λειτουργία της επιχείρησης.

- Ένα πρώτο ενδιαφέρον ερώτημα που μπορεί να εκτελέσει ένας χρήστης αυτής της εφαρμογής είναι να βρει ποια προϊόντα συμμετέχουν λιγότερο στις συναλλαγές (από 0 έως 9) σε κάθε πρατήριο με σκοπό να αποφασίσει αν θέλει να τα αποσύρει τελείως από το συγκεκριμένο πρατήριο ή ακόμα με ένα επιπλέον ερώτημα να βρει ποιος προμηθευτής του το προσέφερε ώστε να διακόψει τη συνεργασία μαζί του.
- Ένα δεύτερο ερώτημα που θα μπορούσε να γίνει είναι να βρούμε σε ποιο πρατήριο γίνονται πολλές συναλλαγές με product τύπου καύσιμο, ώστε να αυξήσουμε τον αριθμό των pipes του πρατηρίου αυτού που προσφέρουν το συγκεκριμένο τύπο για να εξυπηρετηθούν τελικά πιο πολλοί πελάτες.
- Ένα τρίτο ερώτημα που θα μπορούσε να γίνει είναι να δούμε τις συναλλαγές τις οποίες πραγματοποιούνται στα πρατήρια και να παρατηρήσουμε αν συμβαίνουν περισσότερες στα πρατήρια με τύπο λειτουργίας self-service ή full-service ώστε ο ιδιοκτήτης να αποφασίσει ποιος τύπος λειτουργίας είναι πιο κερδοφόρος.
- Ακόμα ένα ερώτημα που θα μπορούσε να γίνει είναι η καταμέτρηση των εγγεγραμμένων πελατών σε ένα συγκεκριμένο πρατήριο και αν ο αριθμός αυτός είναι μικρός (από 0 έως 3) να αυξήσουμε τα corresponding points που αντιστοιχούν σε κάθε προϊόν στο συγκεκριμένο πρατήριο.
- Τελευταίο ερώτημα, να ξέρει ο ιδιοκτήτης ανά πάσας στιγμή ποιο προϊόν του έχει τελειώσει και αν θέλει ειδικότερα να γνωρίζει ποια αντλία έχει αδειάσει για να την γεμίσει.

## 3 ΑΞΙΟΛΟΓΗΣΗ

Η αξιολόγηση του project μας έγινε με βάση κάποια συγκεκριμένα κριτήρια επιτυχίας. Κάποια από αυτά είναι:

- Να συμφωνούν τα δεδομένα της βάσης μας με τα δεδομένα μιας κανονικής αλυσίδας πρατηρίων, να είναι επίκαιρα, έγκυρα και λογικά.
- Να εκτελεί τις βασικές ενέργειες για κάθε table, όπως δημιουργία, εύρεση, ανανέωση και διαγραφή δεδομένων, καλύπτοντας το 1<sup>ο</sup> κριτήριο επιτυχίας.
- Να εκτελεί τα περίπλοκα ερωτήματα που μπορεί να θέσει ο χρήστης.
- Να είναι εύκολο στην χρήση, σχετικά απλό, κατανοητό και ανταγωνιστικό.
- Να είναι φιλικό προς τον χρήστη, ώστε όταν συμπληρώνει ορισμένες φόρμες για την εισαγωγή ή την ανανέωση δεδομένων της επιχείρησης να του εμφανίζονται τα κατάλληλα μηνύματα προτροπής, λάθος ή επιτυχίας.

## 4 ΔΕΔΟΜΕΝΑ

Αποφασίσαμε να δημιουργήσουμε τυχαία δεδομένα για την βάση μας, για να ελέγξουμε αν το project θα μπορούσε να λειτουργήσει σε μεγάλο όγκο πληροφοριών και για να δημιουργήσουμε έπειτα queries για την λήψη στατιστικών δεδομένων. Συμβουλευτήκαμε τις ιστοσελίδες της ΕΚΟ [6] και της SHELL [9] για πληροφορίες πάνω στα καύσιμα [7].

ειδικά προϊόντα [8], προϊόντα minimarket [10] και υπηρεσίες και επιπλέον επισκεφθήκαμε ένα πρατήριο της br για περαιτέρω απορίες που είχαμε όπως:

- Ποσότητα των δεξαμενών
- Πως γίνεται η διαρρύθμιση των δεξαμενών και των αντλιών
- Πως γίνεται η καταγραφή των προμηθειών και κάθε πότε προμηθεύεται το κάθε πρατήριο
- Τα κόστη των καυσίμων, προϊόντων (ειδικών και minimarket) και των υπηρεσιών

και στα υπόλοιπα δεδομένα καταλήξαμε από προσωπική μας κρίση. Έπειτα μέσω του Mockaroo [3] δημιουργήσαμε τυχαία δεδομένα, αρκετά σε πλήθος και έγκυρα με τις πληροφορίες που είχαμε συλλέξει. Τα εξαγάγαμε σε μορφή csv και έπειτα τα ανοίξαμε και τα εισαγάγαμε στην βάση μέσω της python κατά αυτόν τον τρόπο (πχ στο table CONSISTS\_OF):

```
def insertFromCsv():
    fileName = "Datasets/consists_of.csv"
    conn = sqlite3.connect("Gas_Station.db")
    with open(fileName, newline='', encoding='utf_8_sig') as csvfile:
        spamreader = csv.DictReader(csvfile)
        for tuple in spamreader:
            insertInto(tuple['Supply_ID'], tuple['Prod_ID'],
                       tuple['Cost'], tuple['Quantity'], conn)
    conn.close()
```

Αυτή η διαδικασία έγινε για όλα τα 18 tables μας.

## 5 ΚΥΡΙΕΣ ΕΝΕΡΓΕΙΕΣ

Καταρχάς η πρώτη κύρια ενέργεια ήταν η δημιουργία του ERD. Σε αυτό δουλέψαμε και οι δύο μαζί αφού ο πρώτα ο καθένας είχε φτιάξει τη δική του υλοποίηση τις ενόσαμε και μετά την ενδιάμεση παρουσίαση της εργασίας διορθώσαμε κάποια κομμάτια και τελειοποιήσαμε το διάγραμμά μας. Έπειτα συμβουλευτήκαμε τον κ.Αβούρη, ο οποίος μας πρότεινε να επισκεφθούμε ένα πρατήριο, όπως και έκανε ο Αλέξανδρος, για να αντλήσουμε τις απαραίτητες πληροφορίες. Έπειτα ο Ιωάννης είχε την ιδέα να παράγουμε τυχαία δεδομένα για την βάση μας μέσω του Mockaroo [3], οπότε χωρίσαμε τα tables στην μέση, δημιουργήσαμε τα δεδομένα και τα επαληθεύσαμε με τους περιορισμούς που έχουμε θέσει. Τέλος για το κομμάτι του κώδικα τις συνεισφορές μπορείτε να τις δείτε αναλυτικά από το repository στο GitHub.

## 6 ΧΡΟΝΟΔΙΑΓΡΑΜΜΑ

Η πρώτη κύρια ενέργεια χωρίστηκε σε κάποια χρονικά διαστήματα. Στο χρονικό διάστημα 28-31 Οκτωβρίου ο καθένας μας έφτιαξε το δικό του ERD. Στις 6 Νοεμβρίου συντάξαμε το αρχικό πρωτότυπο συγχωνεύοντας τα 2 ERD και πραγματοποιώντας μερικές αλλαγές. Στις 7 Νοεμβρίου προετοιμαστήκαμε για να παρουσιάσουμε την εργασία μας στις 8 Νοεμβρίου. Έπειτα από τις αλλαγές που συζητήσαμε με τους καθηγητές εκείνη την ημέρα τις ενσωματώσαμε στο διάγραμμά μας στις 17 Νοεμβρίου. Έπειτα μέχρι τις 19 Δεκεμβρίου δημιουργήσαμε τυχαία δεδομένα για την βάση μέσω του mockaroo και τα επαληθεύσαμε, ώστε όλα τα tables να συμφωνούν μεταξύ τους. Στις 28 Νοεμβρίου μεσολάβησε η επίσκεψη μας σε ένα πρατήριο της br για περισσότερες πληροφορίες. Στις 19 Δεκεμβρίου δημιουργήσαμε το repository της εργασίας μας στο GitHub. Ξεκινήσαμε με την δημιουργία αρχείων σε python για κάθε table, έπειτα δημιουργήσαμε τα αντίστοιχα html αρχεία και τέλος συμπληρώσαμε τα αρχεία urls.py, με όλα τα urls των tables, και views.py, με τις απαραίτητες κλήσεις συναρτήσεων από τα tables και την μεταφορά των δεδομένων στα αρχεία html, ώστε να προβληθούν. Αυτό εξελίχθηκε μέχρι τις 25 Δεκεμβρίου. Μετά ασχοληθήκαμε κυρίως με τη δημιουργία των βασικών ενεργειών κάθε

table - CRUD (create, read, update, delete) μέχρι τις 4 Ιανουαρίου και τροποποιήσαμε τα αντίστοιχα html αρχεία. Στο χρονικό διάστημα 4-8 Ιανουαρίου έγιναν οι απαραίτητοι περιορισμοί όταν πραγματοποιείται μια βασική ενέργεια, για να συμφωνεί η βάση μας με τους περιορισμούς που έχουμε θέσει στο Σχεσιακό Μοντέλο και που ισχύουν γενικά σε μια αλυσίδα πρατηρίων. Στις 7 Ιανουαρίου δημιουργήσαμε ένα Balance Sheet, στο οποίο φαίνονται τα κέρδη κάθε πρατηρίου σε σχέση με τα προϊόντα/υπηρεσίες και επίσης σε ξεχωριστές ενότητες φαίνονται οι προτιμήσεις που έχουν οι πελάτες σε συγκεκριμένα προϊόντα/υπηρεσίες. Στις 8 Ιανουαρίου ασχοληθήκαμε με την αναφορά και την τελική παρουσίαση της εργασίας μας.

## REFERENCES

- [1] [https://en.wikipedia.org/wiki/Filling\\_station](https://en.wikipedia.org/wiki/Filling_station)
- [2] <https://github.com/>
- [3] <https://www.mockaroo.com/>
- [4] <https://erdmaker.com/>
- [5] <https://www.dbdesigner.net/>
- [6] <https://www.eko.gr/>
- [7] <https://www.eko.gr/pratiria/kafsima-lipantika/>
- [8] <https://www.eko.gr/pratiria/aidika-proionta/>
- [9] <https://www.allsmart.gr/pratiria-shell/shetika-me-ta-pratiria/>
- [10] <https://www.allsmart.gr/pratiria-shell/upiresies-pratirion/smart-shop/>
- [11] <https://www.freepik.com/vectors/gas-pump>

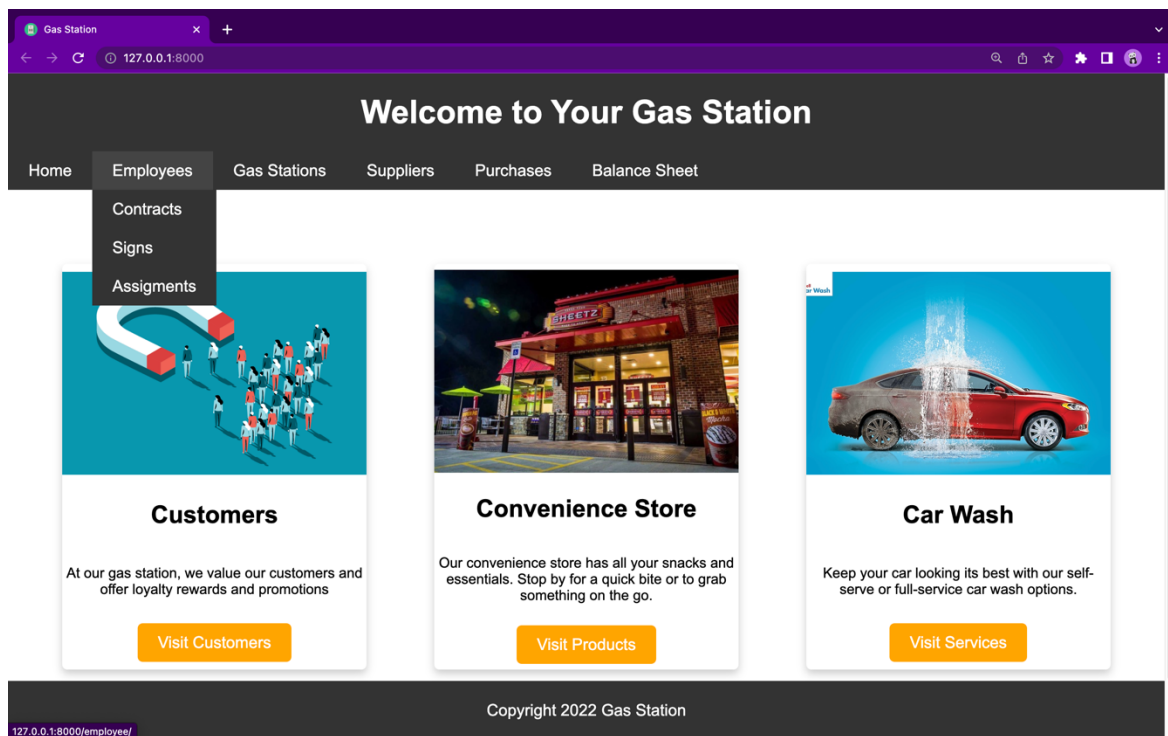
## A ΠΑΡΑΡΤΗΜΑΤΑ

### A.1 Οδηγίες Εγκατάστασης

1. Πήγαινε στον σύνδεσμο [https://github.com/johnnikolaou/Gas\\_Station\\_db](https://github.com/johnnikolaou/Gas_Station_db)
2. Κάνε clone το repository τρέχοντας στο Terminal την εντολή `git clone https://github.com/johnnikolaou/Gas_Station_db.git`
3. Αφού έχεις κατεβάσει το project, για να τρέξεις την εφαρμογή πρέπει να έχεις κατεβασμένη έκδοση της Python 3.6 και πάνω.
4. Για να κατεβάσεις τα απαραίτητα dependencies του project, με προϋπόθεση ότι ο φάκελος που υπάρχει το πρόγραμμα `pip.exe` βρίσκεται στο path σου, τρέξε την εντολή `pip install -r requirements.txt`
5. Τελευταίο βήμα για να αρχίσει η εφαρμογή είναι, με προϋπόθεση ότι ο φάκελος που υπάρχει το πρόγραμμα `python.exe` βρίσκεται στο path σου, να τρέξεις την εντολή `python manage.py runserver`.
6. Αναμένοντας λίγη ώρα για να δημιουργηθεί η βάση και να εισαχθούν τα δεδομένα μπορείς να έχεις πρόσβαση στην ιστοσελίδα γράφοντας το link <http://127.0.0.1:8000/> στο browser ή κάνοντας hover στο link και πατώντας CTRL + click.

### A.2 ΠΑΡΑΔΕΙΓΜΑΤΑ ΧΡΗΣΗΣ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ

Στην αρχική σελίδα προβάλλονται όλες οι επιλογές που έχει ο χρήστης στην διάθεση του.



Μπορεί να αναζητήσει αγορές, προμηθευτές, εργαζόμενους, στατιστικά δεδομένα του κάθε πρατηρίου και πολλά άλλα. Θα δούμε για παράδειγμα τις αγορές (Purchases)

Purchase Handling								
Home	Purchases			Search Purchase		Add Purchase		
Purchase Id	Purchase Date	Type of Payment	Customer Email	Gas Station Longitude	Gas Station Latitude	Pump Id	Tank Id	Actions
1	31/10/22	Cash		38.2019	21.428285	1	1	<a href="#">Edit</a>   <a href="#">Delete</a>
2	01/04/22	Cash	adannatt6t@hud.gov	38.262807	22.778594	1	1	<a href="#">Edit</a>   <a href="#">Delete</a>
3	09/07/22	Debit Card		38.682971	22.038279	1	1	<a href="#">Edit</a>   <a href="#">Delete</a>
4	26/06/22	Credit Card	rbeesoy@ucsd.edu	37.942487	21.152171	1	1	<a href="#">Edit</a>   <a href="#">Delete</a>
5	11/05/22	Debit Card	lsetter8m@printfriendly.com	37.988127	22.115035	1	1	<a href="#">Edit</a>   <a href="#">Delete</a>
6	17/05/22	Cash	osoaper2i@cyberchimps.com	37.793219	22.508819	1	1	<a href="#">Edit</a>   <a href="#">Delete</a>
7	06/01/22	Debit Card	nionnisian3f@oakley.com	36.655448	22.759316	1	1	<a href="#">Edit</a>   <a href="#">Delete</a>
8	13/05/22	Debit Card	rtouseyns@prlog.org	38.007562	21.225408	1	1	<a href="#">Edit</a>   <a href="#">Delete</a>
9	13/07/22	Credit Card	alubecv@shutterfly.com	37.066891	21.200989	1	1	<a href="#">Edit</a>   <a href="#">Delete</a>

Παρατηρούμε ότι εμφανίζονται όλες οι αγορές που έχουν γίνει μέχρι στιγμής. Έπειτα ο χρήστης έχει 4 επιλογές στην διάθεση του, δημιουργία, εύρεση, ανανέωση και διαγραφή αγοράς.

Η διαγραφή είναι η πιο απλή διαδικασία, διότι απλά πατώντας το κουμπί Delete διαγράφεται η αγορά που επιλέχθηκε.

Η ανανέωση σου επιτρέπει να αλλάξεις κάποια τιμή και το αποθηκεύσεις στην βάση με το κουμπί Save. Πρέπει να προσέξουμε, τους περιορισμούς που έχουμε θέσει σε κάθε πεδίο και ότι σε μερικά πεδία δεν μπορούμε να αλλάξουμε τις τιμές τους.

Purchase Date	Type of Payment	Customer Email	Gas Station Longitude	Gas Station Latitude	Pump Id	Tank Id	Actions
/μμ/εεεε <input type="checkbox"/>	Cash		38,2019	21,428285	1	1	<a href="#">Save</a>

Η αναζήτηση σου δίνει την δυνατότητα να αναζητήσεις κάποια αγορά με βάση το Id, την ημερομηνία που πραγματοποιήθηκε, τον τρόπο πληρωμής και πολλά άλλα. Επίσης και εδώ πρέπει να προσέχουμε για τυχόν περιορισμούς.

## Purchase Handling

[Home](#)[Purchases](#)[Search Purchase](#)[Add Purchase](#)

Select fields you want to search for

- Purchase\_Id
- ✓ Purchase\_Date
- Type\_Of\_Payment
- Customer\_Email
- Gas\_Station\_Longitude
- Gas\_Station\_Latitude
- Pump\_Id
- Tank\_Id

purchase-id

submit

Για παράδειγμα κάνουμε αναζήτηση της 13<sup>ης</sup> αγοράς και μπορούμε να την τροποποιήσουμε όπως αναφέραμε παραπάνω.

Home		Purchases		Search Purchase			Add Purchase	
Purchase Id	Purchase Date	Type of Payment	Customer Email	Gas Station Longitude	Gas Station Latitude	Pump Id	Tank Id	Actions
13	26/01/22	Cash	nniblett77@1688.com	36.02151	22.480999	1	1	<a href="#">Edit</a>   <a href="#">Delete</a>

Η δημιουργία αγοράς γίνεται κατά αυτόν τον τρόπο:

Εμφανίζονται τα κουτιά που θα συμπληρώσουμε τις απαραίτητες πληροφορίες για την αγορά. Επιπλέον έχουμε προγραμματίσει έτσι το πρόγραμμα, ώστε να καταλαβαίνει τότε ηλεκτρολογεί ο χρήστης όνομα, email, ημερομηνία κλπ. Αν παρατηρήσουμε προσεκτικά η αγορά παίρνει αυτόματα το Purchase ID 736, διότι δεν μπορούμε να κάνουμε overwrite κάποια άλλη αγορά, οπότε αφού έχουμε 735 αγорές, η επόμενη αγορά θα είναι αυτόματα η 736<sup>η</sup>. Τέλος προσέχουμε και εδώ για τυχόν περιορισμούς που έχουμε θέσει και την ανεβάζουμε στην βάση μέσω του κουμπιού Submit.

Οι παραπάνω ενέργειες ισχύουν για όλα τις επιλογές της αρχική σελίδας, εκτός των στατιστικών δεδομένων (Balance Sheet).

Στο Balance Sheet ο χρήστης έχει την δυνατότητα να δει περαιτέρω πληροφορίες:

- Για κάθε πρατήριο ξεχωριστά, τα κέρδη από τα προϊόντα/υπηρεσίες του και τα κόστη των προϊόντων συνολικά των προμηθειών για το συγκεκριμένο πρατήριο.

Purchase Id:

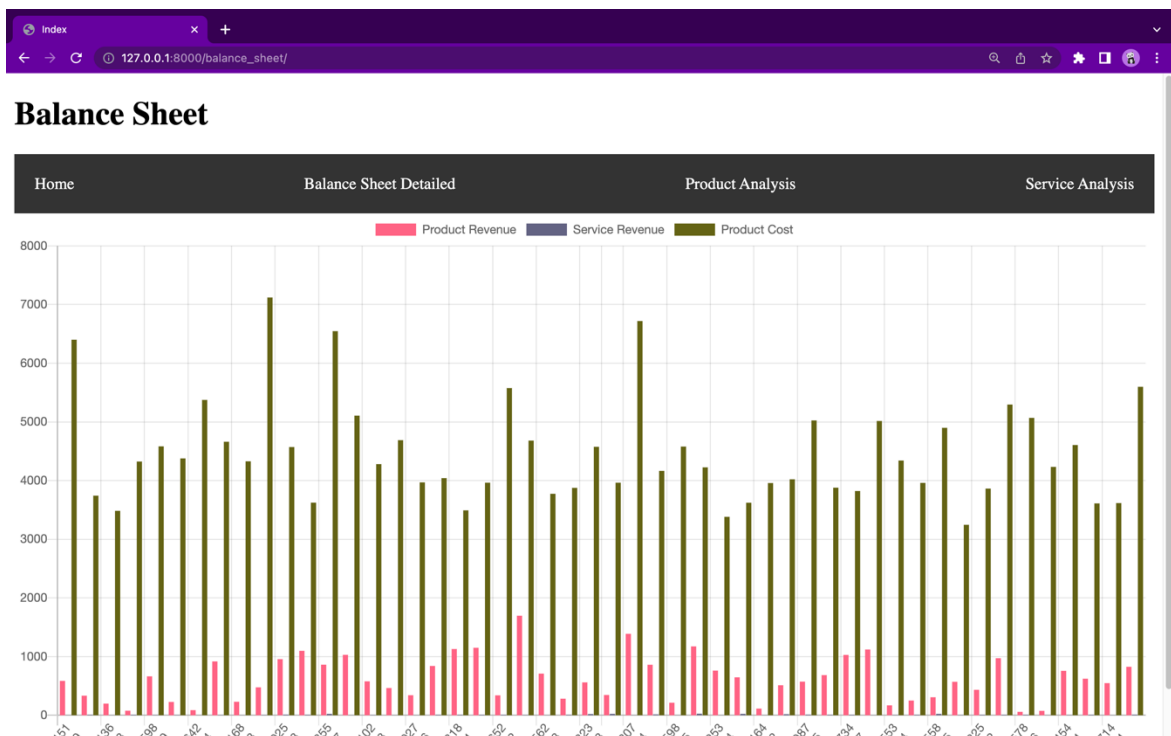
736

Purchase Date:

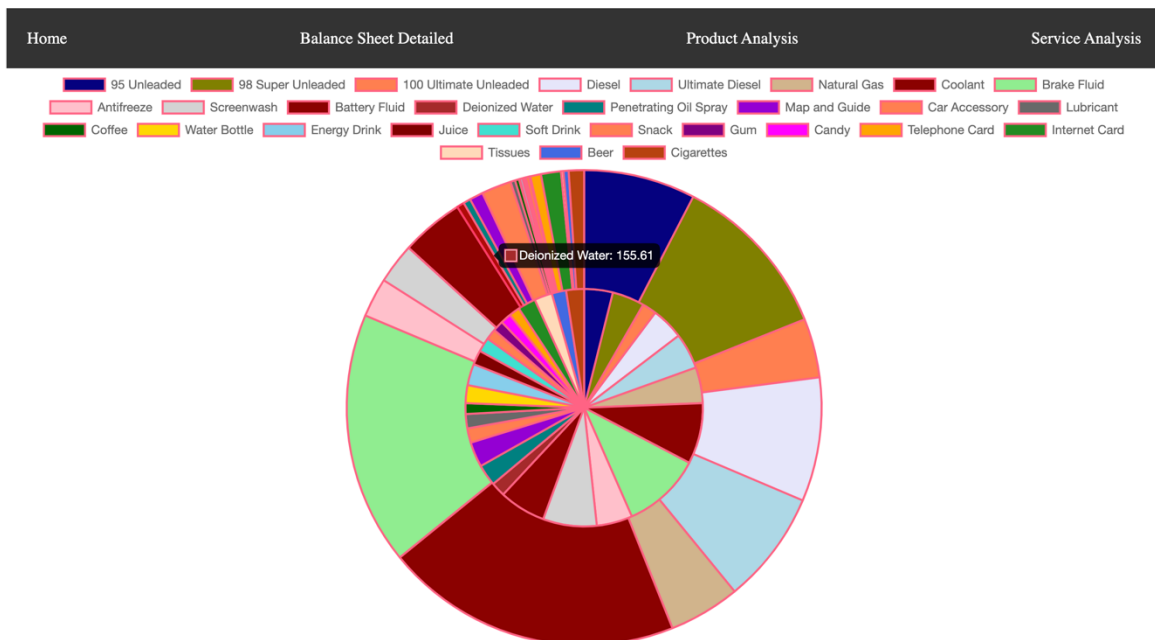
ηη / μμ / εεεε

Type of Payment:

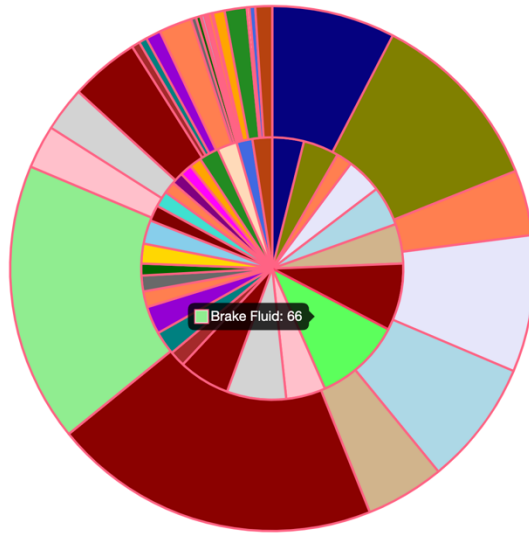
Customer Email:



- Για όλα τα πρατήρια, το εξωτερικό είναι το κέρδος που απέφερε κάθε προϊόν/υπηρεσία και το εσωτερικό είναι η προτίμηση που είχαν οι πελάτες για κάθε προϊόν/υπηρεσία.







Full Wash

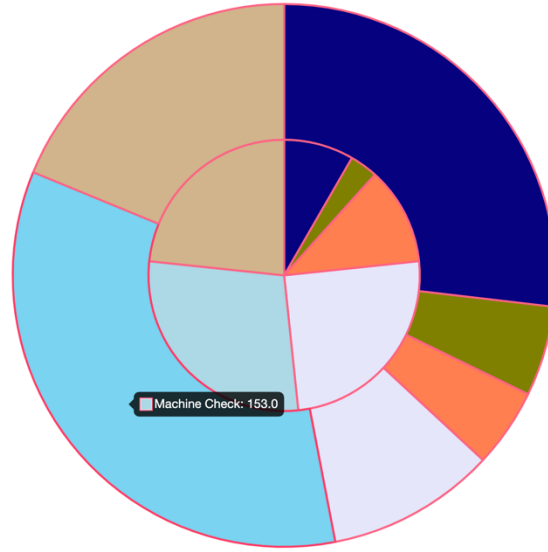
Inside Wash

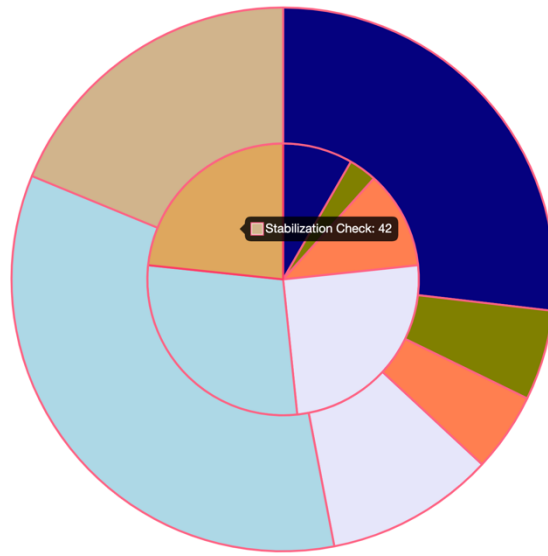
Window Wash

Tire Check

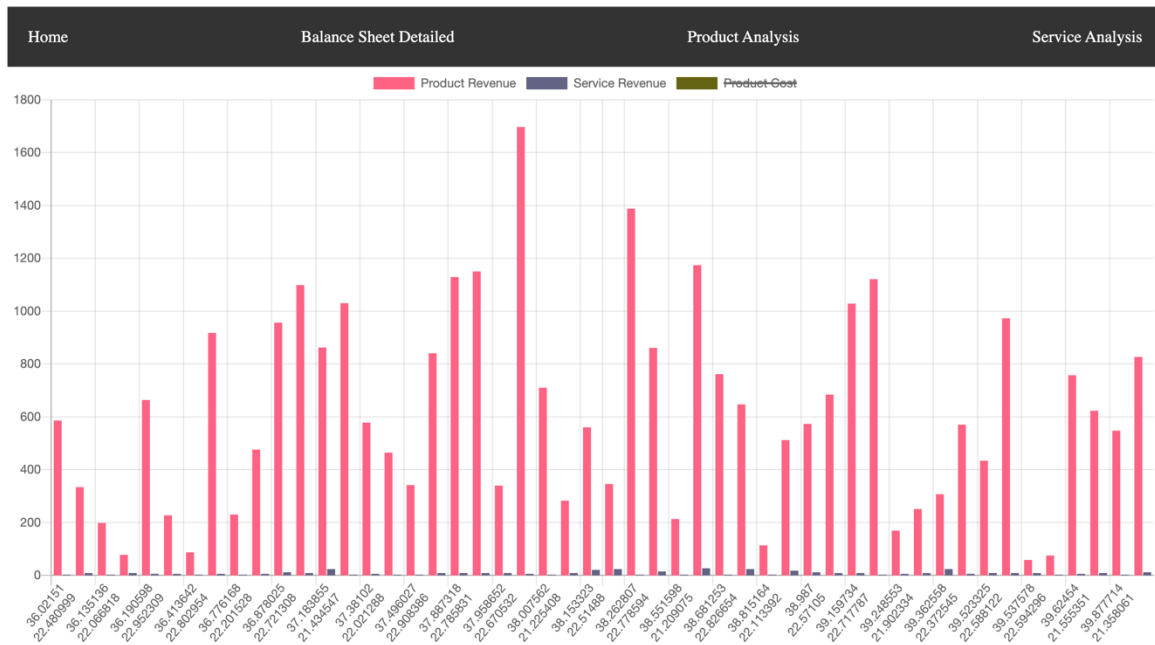
Machine Check

Stabilization Check





Επιπλέον ο χρήστης μπορεί πατώντας κάποια από τις επιλογές πάνω από διαγράμματα μπάρας (bar charts) και πίτας (pie charts) να αφαιρέσει από αυτά τις αντίστοιχες επιλογές, για παράδειγμα αν κάνει κλικ πάνω στο Product Cost θα το αφαιρέσει από το διάγραμμα όπως φαίνεται παρακάτω.



### A.3 ΣΥΝΔΕΣΜΟΣ ΓΙΑ ΤΟΝ ΚΩΔΙΚΑ

[https://github.com/johnnikolaou/Gas\\_Station\\_db](https://github.com/johnnikolaou/Gas_Station_db)