

Ανάπτυξη Εφαρμογής Επεξεργασίας Οικονομικών Μεγεθών

Σπουδαστής: Μπούγας Αθανάσιος-Προκόπιος
Επιβλέπων καθηγητής: Νικολαΐδης Βασίλειος

Καλαμάτα 2016

Κατάλογος περιεχομένων

| | |
|--|----|
| 1. Περίληψη / Abstract | 3 |
| 2. Προεργασία | 4 |
| 2.1 Αντικείμενο Υλοποίησης..... | 4 |
| 2.1.1. Σενάριο χρήσης συστήματος – Use Case | 5 |
| 2.2 Τεχνολογίες και Αρχιτεκτονική | 6 |
| 2.2.1 Γλώσσα προγραμματισμού | 6 |
| 2.2.2 Βάση δεδομένων | 10 |
| 2.2.3 Διεπαφή με το χρήστη | 13 |
| 2.3. Σχεδιασμός..... | 18 |
| 2.3.1 Σχεδιασμός βάσης..... | 18 |
| 2.3.2 Σχεδιασμός με μοντέλα Django | 23 |
| 2.3.3 Σχεδιασμός διεπαφής χρήστη | 24 |
| 2.4. Δείκτες | 26 |
| 3. Εργασία | 28 |
| 3.1. Δομή Κώδικα | 28 |
| 3.2. Βιβλιοθήκη Οικονομικών Συναρτήσεων | 30 |
| 3.3. Templates..... | 33 |
| 3.4. Υπόλοιπα Python αρχεία στο φάκελο economics | 35 |
| 3.5. Υπόλοιπα Python αρχεία στο φάκελο EconomicsWebsite..... | 39 |
| 3.6. Εγκατάσταση Εφαρμογής..... | 41 |
| 3.6.1. Εγκατάσταση MySQL | 41 |
| 3.6.2. Εγκατάσταση και αρχικοποίηση εφαρμογής | 44 |
| 4. Μελλοντική εργασία και επεκτάσεις | 46 |
| Παράρτημα Α: Τελικός σχεδιασμός μοντέλων Django | 47 |
| Παράρτημα Β: SQL-βάση μοντέλων Django | 54 |
| Παράρτημα Γ: Όψη οθονών εφαρμογής | 60 |
| Βιβλιογραφία | 65 |

1. Περίληψη / Abstract

Στην πτυχιακή αυτή, όπως υποδεικνύει και ο τίτλος, αναπτύχθηκε μια εφαρμογή επεξεργασίας των οικονομικών μεγεθών μιας επιχείρησης. Πιο συγκεκριμένα, αναπτύχθηκε μια διαδικτυακή εφαρμογή, στην οποία ο χρήστης μπορεί να αποθηκεύσει προφίλ εταιριών, με τα οικονομικά τους δεδομένα και με βάση αυτά να παράγει αναφορές με τα επεξεργασμένα οικονομικά μεγέθη τους. Σε αυτό το συνοδευτικό κείμενο θα αναλύσουμε:

- Προεργασία: Τι ενέργειες έγιναν για την επιλογή γλώσσας προγραμματισμού, κατάλληλου framework και λοιπών τεχνολογικών επιλογών. Επίσης όλες οι αρχικές ρυθμίσεις που έγιναν με βάση τις επιλογές αυτές (κατασκευή βάσης δεδομένων κ.α.).
- Δείκτες: Ποιους δείκτες υλοποιήσαμε στην εφαρμογή μας.
- Ανάλυση του κώδικα: Θα παρατεθεί ο κώδικας της εφαρμογής και θα αναλυθούν σημεία που χρήζουν ιδιαίτερης προσοχής.
- Μελλοντικοί στόχοι και επεκτάσεις: θα αναλυθεί με ποιος τρόπο η υπάρχουσα εφαρμογή μπορεί να επεκταθεί και να καλύψει περισσότερες/διαφορετικές ανάγκες.

Λέξεις Κλειδιά – keywords:

οικονομικοί δείκτες, διαδικτυακή εφαρμογή, επεξεργασία στοιχείων, οικονομία, εφαρμογή

2. Προεργασία

Όπως ήδη αναφέρθηκε, στον τομέα αυτό θα περιγραφούν οι σκέψεις και οι ενέργειες που έγιναν και μας οδήγησαν στην επιλογή πλατφόρμας, γλώσσας προγραμματισμού, framework υλοποίησης, καθώς και της αρχιτεκτονικής του συστήματος, εφόσον η επιλογή του ενός επηρεάζει άμεσα το άλλο (εννοώντας ότι συγκεκριμένες τεχνολογίες υποστηρίζουν συγκεκριμένες αρχιτεκτονικές). Τέθηκαν υπό σκέψη διάφορες επιλογές, όπως θα δούμε παρακάτω, από τις οποίες τελικά επιλέχθηκε μία τεχνολογία frontend, η οποία και οδήγησε την αρχιτεκτονική και του backend όπως θα δούμε και στη συνέχεια.

2.1 Αντικείμενο Υλοποίησης

Το αντικείμενο της υλοποίησης είναι ένα μια εφαρμογή η οποία θα δίνει τη δυνατότητα στο χρήστη να εισάγει τα οικονομικά στοιχεία μιας εταιρίας και αυτή θα παράγει (κατά απαίτησή του) μια αναφορά με τα επεξεργασμένα στοιχεία αυτής.

Δηλαδή, θα πρέπει να υπάρχει κατ' ελάχιστο μια φόρμα εισαγωγής στοιχείων, η οποία σαν αποτέλεσμα της υποβολής της θα μας δίνει μια σελίδα με υπολογισμένα όλα τα οικονομικά στοιχεία που ζητούνται.

Τα στοιχεία αυτά μας έχουν δοθεί εμμέσως μέσω των τύπων που πρέπει να υπολογίσουμε. Το μόνο που απομένει είναι αποφάσεις για επιπλέον λειτουργικότητα και οι λεπτομέρειες της υλοποίησης από προγραμματιστική σκοπιά.

2.1.1. Σενάριο χρήσης συστήματος – Use Case

A1. Ο χρήστης προσπελαύνει την εφαρμογή

A2. Ο χρήστης εισάγει δεδομένα στην εφαρμογή

A3. Ο χρήστης υποβάλλει την εφαρμογή

A3.1.: αν υπήρξε κάποιο λάθος στα εισηγμένα στοιχεία: επανεμφανίζεται η φόρμα στο χρήστη με το λάθος σημειωμένο. Ο χρήστης καλείται να επαναλάβει τη διαδικασία από το βήμα A2.

A4. Εμφανίζεται στο χρήστη το αποτέλεσμα της επεξεργασίας.

Κείμενο 1: Χαρακτηριστικό Use Case εφαρμογής

Επιπλέον, ο χρήστης θα έχει τη δυνατότητα να διαχειριστεί τα δεδομένα και τις αναφορές τους: διαγραφή, εισαγωγή νέων, κ.α.

2.2 Τεχνολογίες και Αρχιτεκτονική

2.2.1 Γλώσσα προγραμματισμού

Η πρώτη και μεγαλύτερη απόφαση που έπρεπε να παρθεί είναι σε ποια γλώσσα προγραμματισμού θα υλοποιηθεί το σύστημα. Οι δύο υποψήφιος επιλογές ήταν οι εξής:

Java

Η γλώσσα Java δημιουργήθηκε το 1995 από την εταιρία Sun (η οποία δεν υπάρχει πλέον και τη γλώσσα έχει αναλάβει η εταιρία Oracle) και είναι μια γλώσσα σύγχρονη (concurrent), βασισμένη σε κλάσεις και αντικειμενοστραφής, παρόλο που περιέχει και μη αντικειμενοστραφείς τύπους (primitives) και έχει σχεδιαστεί ώστε να έχει όσο το δυνατόν λιγότερες εξαρτήσεις από το υποκείμενο λειτουργικό σύστημα. Το αρχικό της μότο ήταν “γράψε μια φορά, τρέξε παντού” (write once, run anywhere - WORA), το οποίο ισχύει μεν, αλλά με κάποιες παραχωρήσεις. Οι οποίες παραχωρήσεις, πηγάζουν από το γεγονός ότι για να τρέξουν τα προγράμματα της γλώσσας δε μεταφράζονται σε γλώσσα μηχανής αλλά σε μια ενδιάμεση μορφή “γλώσσας μηχανής Java” (bytecode) που τρέχει σε μια Εικονική Μηχανή Java (Java Virtual Machine – JVM).



Εικόνα 1: Λογότυπο της γλώσσας Java

Από το 2014 και μετά είναι μια από τις πιο δημοφιλείς γλώσσες προγραμματισμού και ειδικά για εφαρμογές πελάτη-εξυπηρετητή (client-server) και διαδικτυακές εφαρμογές (web applications), όπου αναφέρεται ότι τη χρησιμοποιούν 9 εκατομμύρια προγραμματιστές παγκοσμίως. Η βασική δομή της γλώσσας έχει επηρεαστεί από τις γλώσσες C/C++ κυρίως, αλλά με πολλές διαφορές και βελτιώσεις πάνω σε αυτές. Από το 2007, η δομή αυτή έχει γίνει δημόσια με την άδεια GNU General Public License (GNU GPL) και έτσι δόθηκε η δυνατότητα σε οποιονδήποτε να φτιάξει το δικό του περιβάλλον Java, αν και το επίσημο της Oracle παραμένει το πιο πλήρες και καθιερωμένο.

Η υποστήριξη της γλώσσας για εφαρμογές διαδικτύου και τη διασύνδεση των συστημάτων είναι ιδιαίτερα αυξημένη, καθώς με την πάροδο των ετών έχει γίνει η de facto επιλογή για τα περισσότερα εταιρικά συστήματα, λόγω των εργαλείων που παρέχει, όπως τα εξής:

- Java Web Services
- JMS ουρές μηνυμάτων
- εταιρικά applets εφαρμογών - EJBs.

Τέλος, όσον αφορά τις γραφικές διεπαφές χρηστών έχουν αναπτυχθεί πάρα πολλά frameworks για την εύκολη, γρήγορη και αποτελεσματική παραγωγή κώδικα, μεταξύ των οποίων τα ακόλουθα:

- Struts/Struts 2
- JSF (Java Server Faces)
- Wicket
- Vaadin
- JSPs & applets

Python

Η γλώσσα Python δημιουργήθηκε το 1991 από τον Guido van Rossum. Είναι μια γλώσσα γενικού σκοπού όπως και η Java, αλλά συγκεντρώνει πολλά είδη προγραμματισμού:

- αντικειμενοστραφές,
- προστακτικό (imperative),
- συναρτησιακό (όχι στο επίπεδο των συναρτησιακών γλωσσών όπως η Lisp, αλλά έχει παραπάνω από αρκετά στοιχεία για να διεκδικήσει και αυτόν τον τίτλο) και
- διαδικαστικό (procedural).



Εικόνα 2: Λογότυπο της γλώσσας Python

Εκτός από το “συλλέκτη σκουπιδιών” (garbage collector) για τη δυναμική αποδέσμευση μνήμης, η γλώσσα έχει και τα εξής πολύ χρήσιμα χαρακτηριστικά/δυνατότητες:

- δυναμικό σύστημα τύπων
- εκτεταμένη βιβλιοθήκη προτύπων (standards library)
- δυναμική επίλυση ονομάτων

Όλα αυτά είναι εφικτά, καθώς η γλώσσα είναι interpreted (δεν παράγεται δηλαδή εκτελέσιμο αρχείο, υπό κανονικές συνθήκες). Αυτό επίσης της δίνει τη δυνατότητα, όπως η Java, να τρέξει σε πολλαπλά συστήματα χωρίς αλλαγές στον κώδικα. Ένα άλλο χαρακτηριστικό της είναι ότι το εύρος των μεταβλητών καθορίζεται από τη στοίχιση των γραμμών και όχι από τη χρήση ειδικών συμβόλων, όπως σε άλλες γλώσσες (“{“ , ”}”).

Η τακτική αυτή είναι εγγενής στη φιλοσοφία της γλώσσας, η οποία συνοψίζεται στο εξής “Zen της Python” (προσβάσιμο στη γλώσσα με την εντολή “import this”):

Επίσης, πρέπει να αναφερθεί ότι η γλώσσα έχει δύο εκδόσεις (branches) αυτή τη στιγμή, που αναπτύσσονται σχεδόν παράλληλα:

- 2.X: με όλα τα σταθερά και δοκιμασμένα χαρακτηριστικά. Αυτή προτιμάται από τους περισσότερους προγραμματιστές.
- 3.X: με όλα τα νέα χαρακτηριστικά και νέες δοκιμές/προσεγγίσεις

Τέλος και η Python παρέχει μια πληθώρα εργαλείων και frameworks για τις διάφορες χρήσεις της γλώσσας, με προεξέχοντα στη διεπαφή χρήστη τα *Django* και *Pyramids*.

Τελική επιλογή γλώσσας προγραμματισμού

Μετά από ώριμη σκέψη και πολλές συγκρίσεις, επιλέχθηκε η γλώσσα Python, για δύο κυρίους λόγους:

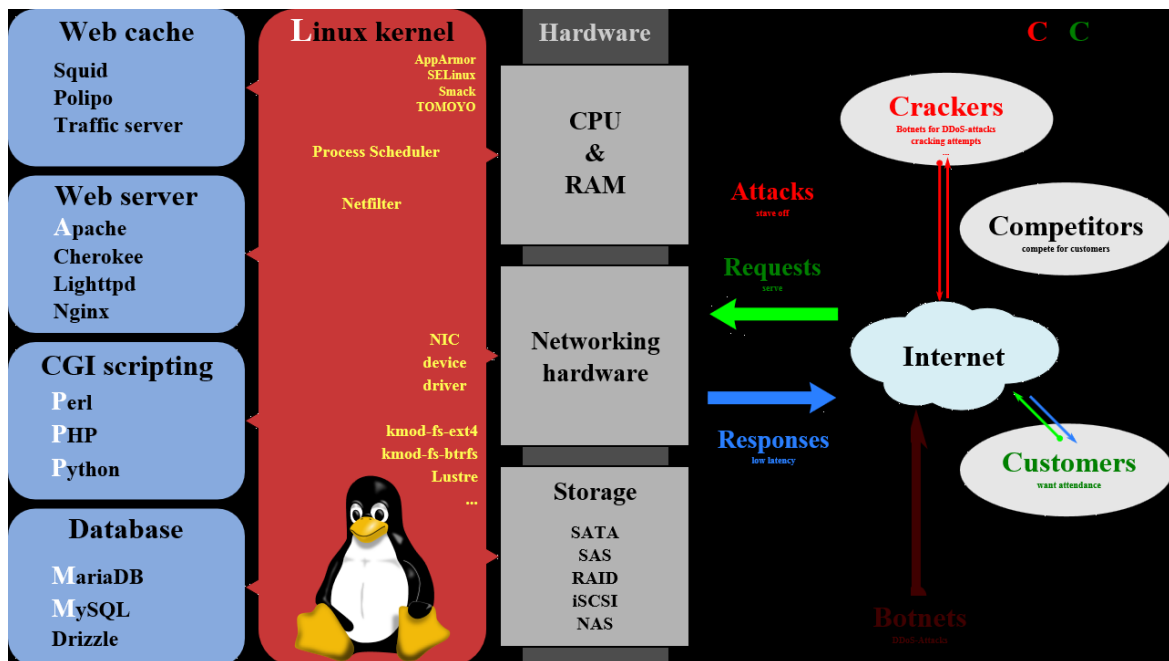
- Δεν υπολείπεται σε τίποτα της γλώσσας Java, παρέχοντας την ίδια λειτουργικότητα στα καίρια σημεία, ενώ ταυτόχρονα:
- Χρησιμοποιείται ευρέως σε ερευνητικά ιδρύματα και πανεπιστήμια σε όλο τον κόσμο.
- Είναι ιδανική για την ανάπτυξη εφαρμογών τέτοιου σκοπού και κλίμακας, καθώς παρέχει πολλές προγραμματιστικές ευκολίες, ακριβώς για τέτοιου είδους εφαρμογές.

2.2.2 Βάση δεδομένων

Η δεύτερη απόφαση είχε να κάνει με τις βάσεις δεδομένων και συγκεκριμένα ποια υλοποίηση θα χρησιμοποιήσουμε. Οι δύο υποψήφιες επιλογές ήταν οι εξής:

MySQL

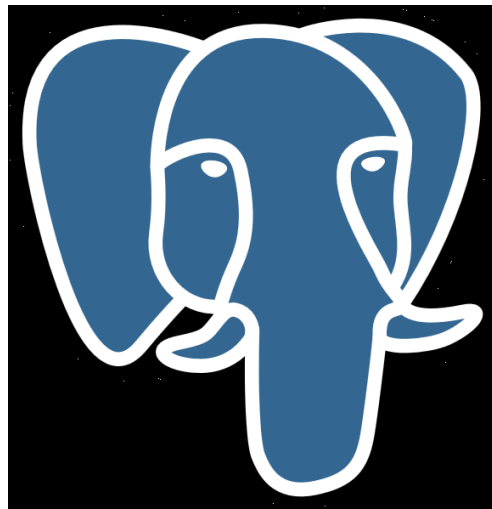
Η βάση δεδομένων MySQL δημιουργήθηκε το 1995 από τη Σουηδική εταιρία *MySQL AB* (η οποία δεν υπάρχει πλέον και τη βάση έχει αναλάβει η εταιρία Oracle) και είναι προϊόν ανοιχτού λογισμικού (open source), αν και έχει και εκδόσεις με έξτρα κλειστό κώδικα, καθώς και εκδόσεις επί πληρωμή. Είναι μια εξαιρετικά δημοφιλής βάση δεδομένων (η 2η πιο δημοφιλής παγκοσμίως από το 2014) για πολλαπλές χρήσεις και παρότι που πολλοί πιστεύουν ότι η χρήση της εξαντλείται στην αρχιτεκτονική LAMP (Linux-Apache-MySQL-PHP, βλέπε εικόνα παρακάτω), το Μάρτιο του 2014 οι εταιρίες Facebook, Google(YouTube), Flickr, Twitter ανακοίνωσαν την υποστήριξη της γλώσσας και την παράλληλη ανάπτυξη σε ανοιχτό λογισμικό, μιας έκδοσής της για τεράστιες ποσότητες δεδομένων (Webscale SQL).



Εικόνα 3: Η αρχιτεκτονική LAMP

PostgreSQL

Η βάση δεδομένων PostgreSQL εκδόθηκε επίσης το 1995, στην Αμερική και αποτελεί την εξέλιξη του έργου Ingres (εξού και το όνομα: post-ingres, μετά την ingres) του πανεπιστημίου του Berkeley, California και είναι επίσης ανοιχτού λογισμικού. Την επίβλεψή της έχει μια μη-κερδοσκοπική εταιρία που έχει συσταθεί από άλλες εταιρίες και δουλεύουν σε αυτοί κυρίως εθελοντές.



Εικόνα 4: Λογότυπο της PostgreSQL

Μέσα στα κυριότερα χαρακτηριστικά της είναι:

- σχεδόν πλήρης συμμόρφωση με τα τελευταία πρότυπα SQL
- αντικειμενοστραφές-σχεσιακό (object-relational) σύστημα βάσης
- πληθώρα τύπου ευρετηρίων
- επεκτασιμότητα

Είναι αρκετά δημοφιλής και αναπτύσσεται συνέχεια, πάντα με μια δικιά της αρκετά ελεύθερη άδεια χρήσης (postgres license).

Τελική επιλογή βάσης δεδομένων

Τελικά επιλέχθηκε η βάση MySQL λόγω της ευελιξίας και της δημοτικότητας της, ιδίως σε έργα τέτοιου μεγέθους. Επίσης, έχει μια σημαντικά πιο μεγάλη και ενεργή κοινότητα χρηστών/διαχειριστών.

2.2.3 Διεπαφή με το χρήστη

Από τη στιγμή που επιλέχθηκε η γλώσσα Python για την υλοποίηση του συστήματος, έπρεπε να γίνει μια επιλογή ανάμεσα στα διάφορα frameworks/libraries που παρέχονται για τη γλώσσα αυτή, για την κατασκευή διεπαφών με το χρήστη. Από τα εξής υπάρχοντα συστήματα:

(<https://wiki.python.org/moin/WebFrameworks>)

- Django
- Grok
- Pylons
- TurboGears
- web2py
- Zope2

επιλέχθηκε το **Django**, ως το πιο δημοφιλές και ενεργά αναπτυσσόμενο, με τη μεγαλύτερη κοινότητα, framework. Επιπλέον, υλοποιεί το σχεδιασμό Model-View-Controller, που είναι δοκιμασμένος και αποτελεσματικός και αναλύεται παρακάτω, αφού παραθέσουμε μερικές επιπλέον πληροφορίες για το Django.

Django

Το **Django** εκδόθηκε αρχικά το 2005 από δυο προγραμματιστές, τους Adrian Holovaty και Simon Willison, με αρχικό σκοπό να κάνει πιο εύκολη την ανάπτυξη πολύπλοκων ιστότοπων με έμφαση στις βάσεις (database-driven sites). Πλέον, το διαχειρίζεται ένα μη κερδοσκοπικό ίδρυμα, το Django Software Foundation και παραμένει ανοικτού κώδικα.



Εικόνα 5: Λογότυπο Django

Το Django δίνει έμφαση στην επαναχρησιμοποίηση συστατικών (component reusability) και την δόμησή τους σε πιο πολύπλοκα συστήματα (pluggability). Μεγάλα του ατού είναι:

- χρησιμοποιεί τη model-view-controller αρχιτεκτονική (στην επόμενη παράγραφο θα αναλυθεί το πώς)
- παρέχει μια αυτοματοποιημένη, αλλά πλήρως παραμετροποιήσιμη, διαχειριστική διεπαφή για τις πιο συχνές λειτουργίες στα μοντέλα (δημιουργία, ανάγνωση, ανανέωση, διαγραφή – CRUD)
- παρέχει έναν ελαφρύ web server με ενσωματωμένη βάση δεδομένων για άμεσο έλεγχο του site
- παρέχει εγγενή υποστήριξη προς τις περισσότερες βάσεις δεδομένων
- παρέχει αυτοματοποιημένη δημιουργία και διαχείριση φορμών
- παρέχει ένα σύστημα ενδιάμεσης μνήμης (cache)

- παρέχει ένα αυτόματο σύστημα για μετατροπή από/προς XML & JSON

Το framework είναι πολύ δημοφιλές και το χρησιμοποιούν μερικά sites μεγάλου βεληγεκούς, όπως:

- Pinterest
- Instagram
- Mozilla
- The Washington Times

Αρχιτεκτονική Model-View-Controller (MVC)

Η αρχιτεκτονική Μοντέλου-Παρουσίασης-Ελεγκτή (Model-view-controller: **MVC**) είναι μια μέθοδος (pattern) ανάπτυξης λογισμικού διεπαφών με χρήστες, που βρήκε πρόσφορα έδαφος κυρίως στην ανάπτυξη εφαρμογών για το διαδίκτυο.

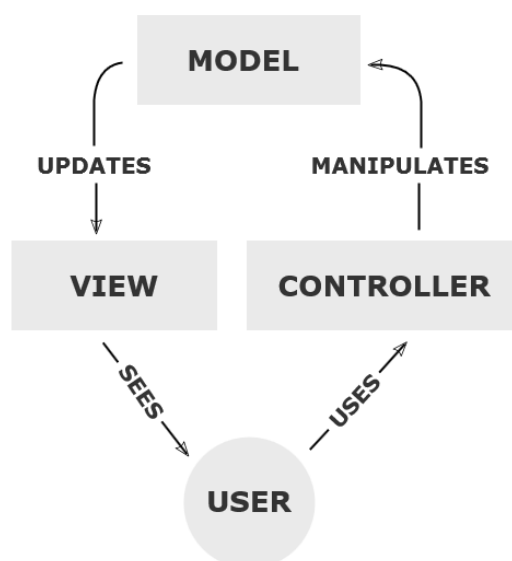
Με βάση αυτό, η εφαρμογή χωρίζεται σε τρία μέρη, για να επιτύχει το διαχωρισμό της εσωτερικής αναπαράστασης της πληροφορίας από τον τρόπο που παρουσιάζεται στο χρήστη:

- Το **Μοντέλο** αποτελείται από τα δεδομένα της εφαρμογής, τους κανόνες, τη λογική και τις τυχόν συναρτήσεις/μεθόδους.
- Η **Παρουσίαση** είναι οποιαδήποτε αναπαράσταση της πληροφορίας που περιέχεται στο μοντέλο με οποιοδήποτε τρόπο (πχ. κείμενο, πίνακας, διάγραμμα, χάρτης)
- Ο **Ελεγκτής** δέχεται ως είσοδο το μοντέλο και το μετατρέπει σε εντολές, κατάλληλες για επίδειξη από την παρουσίαση. Πολλές φορές οι κανόνες και η λογική του προγράμματος βρίσκονται εδώ, παρά στο μοντέλο.

Οι αλληλεπιδράσεις μεταξύ των τριών αυτών συνιστωσών είναι καλά καθορισμένες:

- Το **Μοντέλο** ενημερώνει (notifies) τις σχετιζόμενες με αυτό Παρουσιάσεις και Ελεγκτές όταν γίνεται αλλαγή στην κατάστασή του. Αυτή η ενημέρωση επιτρέπει στις Παρουσιάσεις να αλλάξουν την οπτικοποίησή τους και στους Ελεγκτές τις εντολές τους (και τη λογική τους, αν περιέχεται σε αυτούς). Υπάρχει όμως και η επιλογή του “Παθητικού Μοντέλου”, το οποίο δεν ενημερώνει ποτέ αλλά αντιθέτως δέχεται “αιτήσεις ενημέρωσης” (polls) και απαντάει σε αυτές.
- Η **Παρουσίαση** λαμβάνει από τον Ελεγκτή τις πληροφορίες που χρειάζεται για την παραγωγή εξόδου κατανοητής από το χρήστη. Επίσης δύναται να έχει μηχανισμούς για τη μεταφορά της εισόδου του χρήστη στον Ελεγκτή.
- Ο **Ελεγκτής**, όπως αναφέρθηκε, μπορεί να στέλνει τόσο αιτήσεις στο Μοντέλο και πληροφορίες στο μοντέλο, όσο και εντολές στην Παρουσίαση.

Όλα αυτά φαίνονται στο παρακάτω σχήμα.



Το Django

συμμορφώνεται με

Εικόνα 6: Μοντέλο-Παρουσίαση-Ελεγκτής το πρότυπο MVC,

ως εξής:

- υποστηρίζει **Μοντέλα**, που είναι κλάσεις Python με τη μοντελοποίηση των δεδομένων σε μία σχεσιακή βάση
- η **Παρουσίαση** είναι ένα σύστημα προτύπων (templates) με τα οποία παράγονται οι σελίδες html (και είναι τα ίδια σελίδες HTML με την προσθήκη μια πρότυπης γλώσσας προτύπων – Django Template Language)
- ο **Ελεγκτής** συνίσταται από αρχεία λογικής Python σε συνδυασμό με ένα σύστημα ανακατεύθυνσης σε URL βάσει προτύπων εκφράσεων (regular expressions)

2.3. Σχεδιασμός

Ο σχεδιασμός του συστήματος έγινε σε δύο ξεχωριστές φάσεις:

1. Το σχεδιασμό της βάσης/των δεδομένων
2. Το σχεδιασμό της εμπειρίας του χρήστη (user interface)

Παρακάτω αναλύονται και οι δύο σχεδιασμοί βήμα-προς-βήμα.

2.3.1 Σχεδιασμός βάσης

Η πρώτη απόφαση ήταν ότι θα χρησιμοποιηθεί η σχεσιακή βάση δεδομένων MySQL (κάτι που αναλύθηκε ήδη στην παράγραφο 2.2.2 *Βάση δεδομένων*), το οποίο είναι καθοριστικό για τη μορφή που θα έχει το μοντέλο δεδομένων.

Αρχικά πρέπει να ορίσουμε τις οντότητες του συστήματος:

1. Για τους χρήστες, θα χρησιμοποιηθεί η έτοιμη οντότητα του Django.
Δε θα προστεθεί νέος πίνακας στη βάση για αυτό το σκοπό.
2. Η βασική οντότητα του συστήματος είναι η **είσοδος δεδομένων του χρήστη**. Αυτή είναι και η βάση παραγωγής των αναφορών. Περιέχει όλα τα δεδομένα που χρειάζονται για να υπολογιστούν οι διάφοροι δείκτες, δηλαδή τα εξής:
 - Μικτά Κέρδη Εκμετάλλευσης
 - Καθαρές Πωλήσεις Χρήσεων
 - Καθαρά Κέρδη Εκμετάλλευσης
 - Χρηματοοικονομικά Έξοδα
 - Συνολικά Απασχολούμενα Κεφάλαια
 - Συνολικό Ενεργητικό
 - Σύνολο Ιδίων Κεφαλαίων
 - Αποδοτικότητα Ιδίων Κεφαλαίων
 - Αποδοτικότητα Συνόλου Απασχολούμενων Κεφαλαίων
 - Καθαρά Κέρδη Προ Φόρων
 - Μ.Ο. Ενεργητικού

- Ιδία Κεφάλαια
- Ξένα Κεφάλαια
- Συνολικά Κεφάλαια
- Πάγια Κεφάλαια
- Μακροχρόνιες Υποχρεώσεις
- Κυκλοφορούν Ενεργητικό
- Σύνολο Υποχρεώσεων
- Σύνολο Τόκων
- Καθαρές Πωλήσεις
- Μέσος Όρος Απαιτήσεων
- Κόστος Πωληθέντων
- Μ.Ο.Β.Υ
- Μέσο Απόθεμα Προϊόντων
- Ενεργητικό Αρχής
- Ενεργητικό Τέλους
- Πάγιο Ενεργητικό
- Σύνολο Καθαρών Κερδών
- Αριθμός Μετοχών
- Σύνολο Μερισμάτων
- Αριθμός Μετοχών σε Κυκλοφορία
- Μέρισμα Ανά Μετοχή
- Τιμή Μετοχής
- Καθαρά Κέρδη Χρήσεων
- Σύνολο Καταβαλλόμενων Μερισμάτων
- Χρηματιστηριακή Τιμή
- Κέρδη Ανά Μετοχή
- Διαθέσιμα
- Απαιτήσεις
- Βραχυπρόθεσμες Υποχρεώσεις

- Διαθέσιμο Ενεργητικό
- Ληξιπρόθεσμες Υποχρεώσεις
- Προβλεπόμενες Ημερήσιες Λειτουργικές Δαπάνες

Και τα παρακάτω που χρειάζονται για τη σωστή αποθήκευση των δεδομένων:

- Όνομα Εταιρίας
- Ημερομηνία Δημιουργίας
- Ημερομηνία Τελευταίας Αλλαγής

Όλα τα παραπάνω έχουν δηλωθεί ως Decimal με δύο δεκαδικά ψηφία ή/και Ημερομηνίες.

3. Η δεύτερη οντότητα είναι η παραγόμενη αναφορά, ή αλλιώς: **η έξοδος από τα δεδομένα του χρήστη**. Μετά από πολύ σκέψη για το πώς θα ενωθεί με την πρώτη οντότητα (ξένο κλειδί, κληρονομικότητα, αντιγραφή δεδομένων), τελικά προτιμήθηκε η λύση της πλήρης αντιγραφής των δεδομένων, με την εξής λογική:

- Η ποσότητα των δεδομένων είναι μικρή
- Η πολυπλοκότητα που προκαλούσαν πολλαπλές εκδόσεις των δεδομένων και των εξόδων τους, καθώς και η διαχείρισή τους, δε δικαιολογούσε την ευκολία που προσφέρει ένα ξένο κλειδί

Τελικά τα πεδία που περιέχονται είναι τα εξής από την είσοδο δεδομένων:

- Μικτά Κέρδη Εκμετάλλευσης
- Καθαρές Πωλήσεις Χρήσεων
- Καθαρά Κέρδη Εκμετάλλευσης
- Χρηματοοικονομικά Έξοδα
- Συνολικά Απασχολούμενα Κεφάλαια
- Συνολικό Ενεργητικό
- Σύνολο Ιδίων Κεφαλαίων

- Αποδοτικότητα Ιδίων Κεφαλαίων
- Αποδοτικότητα Συνόλου Απασχολούμενων Κεφαλαίων
- Καθαρά Κέρδη Προ Φόρων
- Μ.Ο. Ενεργητικού
- Ιδία Κεφάλαια
- Ξένα Κεφάλαια
- Συνολικά Κεφάλαια
- Πάγια Κεφάλαια
- Μακροχρόνιες Υποχρεώσεις
- Κυκλοφορούν Ενεργητικό
- Σύνολο Υποχρεώσεων
- Σύνολο Τόκων
- Καθαρές Πωλήσεις
- Μέσος Όρος Απαιτήσεων
- Κόστος Πωληθέντων
- Μ.Ο.Β.Υ
- Μέσο Απόθεμα Προϊόντων
- Ενεργητικό Αρχής
- Ενεργητικό Τέλους
- Πάγιο Ενεργητικό
- Σύνολο Καθαρών Κερδών
- Αριθμός Μετοχών
- Σύνολο Μερισμάτων
- Αριθμός Μετοχών σε Κυκλοφορία
- Μέρισμα Ανά Μετοχή
- Τιμή Μετοχής
- Καθαρά Κέρδη Χρήσεων
- Σύνολο Καταβαλλόμενων Μερισμάτων
- Χρηματιστηριακή Τιμή

- Κέρδη Ανά Μετοχή
- Διαθέσιμα
- Απαιτήσεις
- Βραχυπρόθεσμες Υποχρεώσεις
- Διαθέσιμο Ενεργητικό
- Ληξιπρόθεσμες Υποχρεώσεις
- Προβλεπόμενες Ημερήσιες Λειτουργικές Δαπάνες
- Όνομα Εταιρίας
- Ημερομηνία Δημιουργίας
- Ημερομηνία Τελευταίας Αλλαγής

Και οι δείκτες-αποτελέσματα:

- Αριθμοδείκτης Μεικτού Περιθωρίου
- Αποδοτικότητα Απασχολούμενων Κεφαλαίων Τιμή Μετοχής
- P.O.A.
- P.O.E.
- Αριθμοδείκτης Οικονομικής Μόχλευσης
- P.O.A. (B τρόπος)
- P.O.I.
- Αριθμοδείκτης Ιδίων Προς Συνολικά Κεφάλαια
- Αριθμοδείκτης Ιδίων Προς Ξένα Κεφάλαια
- Αριθμοδείκτης Ιδίων Προς Πάγια Κεφάλαια
- Αριθμοδείκτης Παγίων Κεφαλαίων Προς Μακροχρόνιες Υποχρεώσεις
- Αριθμοδείκτης Κυκλοφορούντος Ενεργητικού Προς Συνολικές Υποχρεώσεις
- Αριθμοδείκτης Κάλυψης Τόκων
- Αριθμοδείκτης Ταχύτητας Είσπραξης Απαιτήσεων
- Αριθμοδείκτης Ταχύτητας Βραχυπρόθεσμων Υποχρεώσεων
- Αριθμοδείκτης Ταχύτητας Κυκλοφορίας Αποθεμάτων

- Μ.Ο. Ενεργητικού
- Αριθμοδείκτης Ταχύτητας Κυκλοφορίας Καθαρού Κεφαλαίου
- Αριθμοδείκτης Ταχύτητας Κυκλοφορίας Ιδίων Κεφαλαίων
- Αριθμοδείκτης Ταχύτητας Κυκλοφορίας Παγίων Κεφαλαίων
- Κέρδη ανά Μετοχή
- Μέρισμα ανά Μετοχή
- Μερισματική Απόδοση
- Αριθμοδείκτης Κάλυψης Καταβαλλόμενων Μερισμάτων
- Εσωτερική Αξία Μετοχής
- Π προς Ε.Π.Σ.
- Αριθμοδείκτης Γενικής Ρευστότητας
- Αριθμοδείκτης Ειδικής Ρευστότητας
- Αριθμοδείκτης Ταμειακής Ρευστότητας
- Αριθμοδείκτης Αμυντικού Χρονικού Διαστήματος

Μαζί με το παρακάτω για την καλύτερη οργάνωση στη διεπαφή:

- Ημερομηνία Αποθήκευσης

Όλα τα παραπάνω έχουν δηλωθεί ως Decimal με δύο δεκαδικά ψηφία ή/και Ημερομηνίες.

2.3.2 Σχεδιασμός με μοντέλα Django

Το επόμενο βήμα ήταν η προσαρμογή του αρχικού σχεδιασμού στο Django. Στο framework αυτό τα μοντέλα και οι μεταξύ τους σχέσεις περιγράφονται σε Python και παράγεται η βάση (SQL) εκ των υστέρων, με βάση το σχεδιασμό αυτό.

Ολόκληρος ο κώδικας του Django φαίνεται στο . Επίσης, στο θα βρείτε το μεγαλύτερο μέρος της παραγόμενης βάσης σε MySQL.

2.3.3 Σχεδιασμός διεπαφής χρήστη

Το τελικό μέρος του σχεδιασμού αφορά τη σχεδίαση της διεπαφής του χρήστη, δηλαδή τις οθόνες-σελίδες που θα βλέπει ο χρήστης και θα αλληλοεπιδρά με το σύστημα. Για το look-and-feel επιλέχθηκε ένα λιτό πλην όμως λειτουργικό theme στις αποχρώσεις του γκρι. Όσον αφορά τις οθόνες, θα υπάρχουν οι εξής αρχικά, με προοπτική προσθήκης νέων οθονών, όσο προχωράει το έργο:

1. *οθόνη εισόδου χρήστη στο σύστημα (username-password)*
 - Στην οθόνη αυτή, ο χρήστης μπορεί να εισάγει όνομα χρήστη και κωδικό και να μπει στο σύστημα.
2. *αρχική οθόνη όπου ο χρήστης θα βλέπει τον αριθμό των Στοιχείων Εισόδου και Αναφορών που υπάρχουν στο σύστημα*
3. *οθόνη επισκόπησης Υπαρχουσών Στοιχείων Εισόδων.* Για κάθε στοιχείο Εισόδου μπορεί να γίνει:
 - Επεξεργασία (με κλικ στον τίτλο του στοιχείου).
 - Διαγραφή
 - Παραγωγή Αναφοράς με τα υπάρχοντα στοιχεία
 - Κλωνοποίηση (αντιγραφή σε νέα οντότητα)
4. *οθόνη δημιουργίας Στοιχείων Εισόδου*
5. *οθόνη επισκόπησης Υπαρχουσών Αναφορών.* Για κάθε Αναφορά μπορεί να γίνει:
 - Προβολή των Στοιχείων Εισόδου.
 - Διαγραφή
 - Προβολή της Αναφοράς (σε νέο παράθυρο, χωρίς μενού, για εκτύπωση)

Μια αρχική όψη της υλοποίησης των οθονών αυτών μπορείτε να δείτε στο .

Επιπλέον οθόνες που υπάρχουν και απορρέουν από τις 5 βασικές είναι:

- *Οθόνη λάθους/404*
 - όπου εμφανίζεται στο χρήστη μήνυμα λάθους / 404, λάθος αιτήματος σελίδας
- *Οθόνη μη επαρκών δικαιωμάτων*
 - όπου εμφανίζεται στο χρήστη μήνυμα ότι δεν έχει επαρκή δικαιώματα για μια ενέργεια
- *Οθόνες μηνυμάτων επιτυχίας*
 - όπου εμφανίζεται στο χρήστη μήνυμα επιτυχίας για κάποια ενέργεια (π.χ. εκκίνηση εργασίας)

Τέλος, πολύ μεγάλης σημασίας είναι οι οθόνες διαχείρισης, οι οποίες παρέχονται από το Django και δίνουν τη δυνατότητα στον χρήστη με τα κατάλληλα δικαιώματα:

- να κάνει επισκόπηση των ήδη υπαρχόντων δεδομένων στο σύστημα, και
- να δημιουργήσει νέες οντότητες και χρήστες,
- να διαγράψει οντότητες με ασφάλεια (cascading deletes).

2.4. Δείκτες

Όπως αναφέρθηκε και στο κεφάλαιο κατασκευής της βάσης, οι Δείκτες οι οποίοι καλούνται να υπολογιστούν είναι οι εξής:

- Αριθμοδείκτης Μεικτού Περιθωρίου
- Αποδοτικότητα Απασχολούμενων Κεφαλαίων Τιμή Μετοχής
- P.O.A. (με 2 τρόπους)
- P.O.I.
- P.O.E.
- Αριθμοδείκτης Οικονομικής Μόχλευσης
- Αριθμοδείκτης Ιδίων Προς Συνολικά Κεφάλαια
- Αριθμοδείκτης Ιδίων Προς Ξένα Κεφάλαια
- Αριθμοδείκτης Ιδίων Προς Πάγια Κεφάλαια
- Αριθμοδείκτης Παγίων Κεφαλαίων Προς Μακροχρόνιες Υποχρεώσεις
- Αριθμοδείκτης Κυκλοφορούντος Ενεργητικού Προς Συνολικές Υποχρεώσεις
- Αριθμοδείκτης Κάλυψης Τόκων
- Αριθμοδείκτης Ταχύτητας Είσπραξης Απαιτήσεων
- Αριθμοδείκτης Ταχύτητας Βραχυπρόθεσμων Υποχρεώσεων
- Αριθμοδείκτης Ταχύτητας Κυκλοφορίας Αποθεμάτων
- M.O. Ενεργητικού
- Αριθμοδείκτης Ταχύτητας Κυκλοφορίας Καθαρού Κεφαλαίου
- Αριθμοδείκτης Ταχύτητας Κυκλοφορίας Ιδίων Κεφαλαίων
- Αριθμοδείκτης Ταχύτητας Κυκλοφορίας Παγίων Κεφαλαίων
- Κέρδη ανά Μετοχή
- Μέρισμα ανά Μετοχή
- Μερισματική Απόδοση
- Αριθμοδείκτης Κάλυψης Καταβαλλόμενων Μερισμάτων

- Εσωτερική Αξία Μετοχής
- Π προς Ε.Π.Σ.
- Αριθμοδείκτης Γενικής Ρευστότητας
- Αριθμοδείκτης Ειδικής Ρευστότητας
- Αριθμοδείκτης Ταμειακής Ρευστότητας
- Αριθμοδείκτης Αμυντικού Χρονικού Διαστήματος

3. Εργασία

Όπως ήδη αναφέρθηκε, στον τομέα αυτό θα ασχοληθούμε με τα πρακτικά ζητήματα: θα αναλυθεί ο κώδικας που γράφτηκε για την εφαρμογή, καθώς και θα γίνει μια περιγραφή της εγκατάστασης και εκτέλεσης της εφαρμογής.

3.1. Δομή Κώδικα

Κατ' αρχήν πρέπει να αναφερθεί ότι ο κώδικας είναι χωρισμένος σε δύο τμήματα:

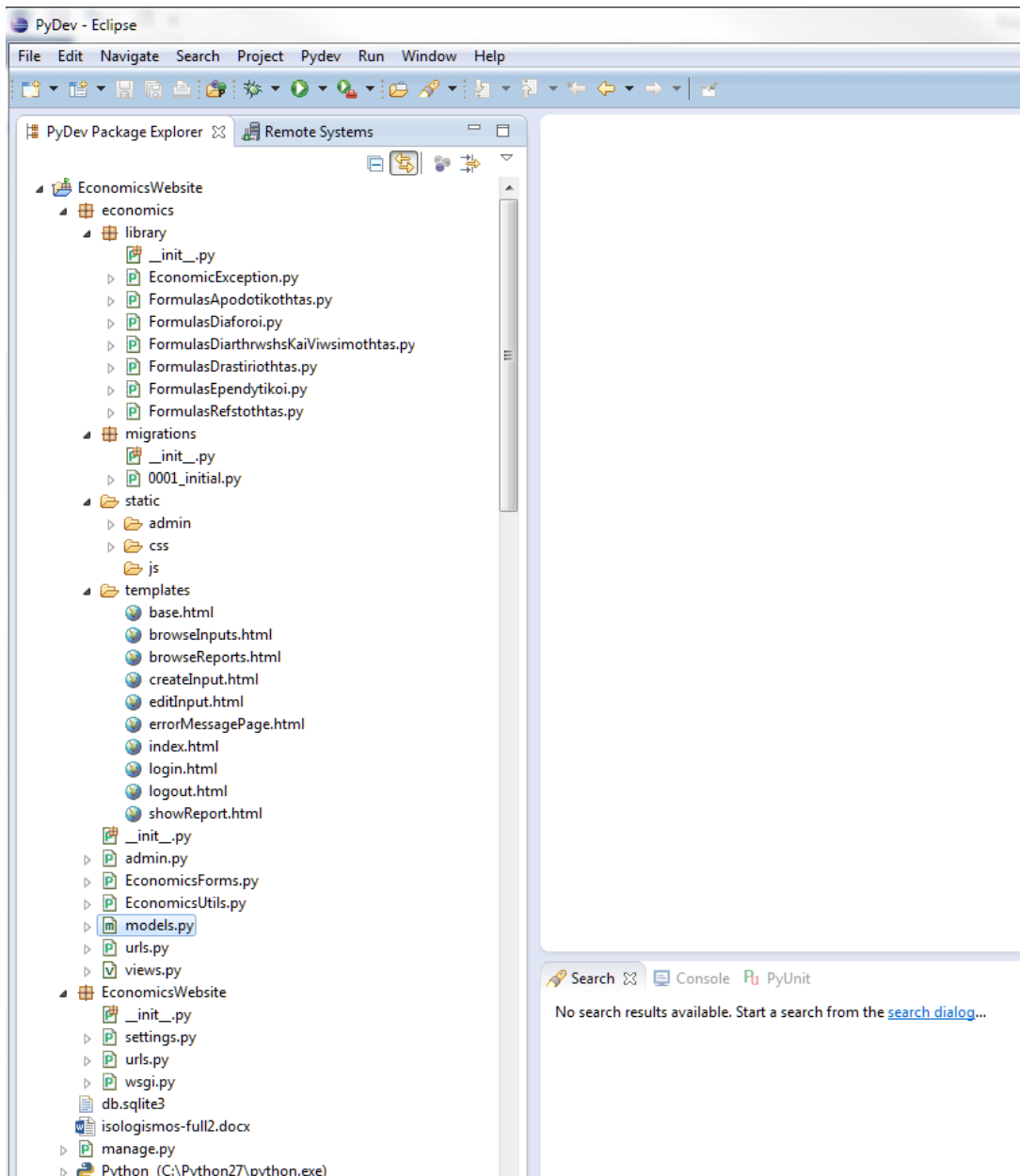
1. το τμήμα της βιβλιοθήκης συναρτήσεων και
2. το τμήμα της γραφικής διεπαφής (GUI – Graphical User Interface).

Τα οποία θα αναλυθούν χωριστά, καθώς δομικά είναι εντελώς ανεξάρτητα (απλώς το δεύτερο χρησιμοποιεί το πρώτο).

Ας ξεκινήσουμε όμως με τα βασικά του κώδικα. Η ανάπτυξη έχει γίνει σε Eclipse με Python extensions, για αυτό υπάρχουν και τα σχετικά αρχεία “.project” κ.τ.λ. Η δομή των αρχείων φαίνεται στην Εικόνα 7.

Όπως είναι προφανές, το έργο (project) είναι χωρισμένο σε καταλόγους. Αναφέρονται οι κυριότεροι, με επεξήγηση του τι περιέχουν.

- 1. Economics:** ο κύριος κατάλογος της εφαρμογής με τα πηγαία αρχεία
- 2. Migrations:** ο κατάλογος στον οποίο αποθηκεύει το Django τα αρχεία αλλαγής βάσης, ώστε να μπορεί να αναδημιουργήσει τη δομή της βάσης εκ του μηδενός.
- 3. Static:** ο κατάλογος με τα JavaScript, CSS και εικόνες (στατικό περιεχόμενο) για τη διεπαφή χρήστη
- 4. Templates:** ο κατάλογος με τις HTML σελίδες από τις οποίες φτιάχνονται οι τελικές σελίδες της εφαρμογής μέσω του Django
- 5. EconomicsWebsite:** τα βασικά Python αρχεία με τα οποία γίνονται οι ρυθμίσεις της εφαρμογής (αντίστοιχο του «web-inf» στην Java).



Εικόνα 7 Δομή αρχείων Κώδικα

3.2. Βιβλιοθήκη Οικονομικών Συναρτήσεων

Η βιβλιοθήκη οικονομικών συναρτήσεων βρίσκεται στον υποκατάλογο **library** του **economics**. Αποτελείται από τα εξής αρχεία:

- **EconomicException.py**

- Περιέχει την εξαίρεση λειτουργίας (exception) που πετάγεται από το πρόγραμμα (raised) σε περίπτωση λάθους.

Σύμφωνα με τις συμβάσεις της Python, ο *EconomicsInvalidOperantError* αποθηκεύει μέσα του το μήνυμα λάθους με το οποίο αρχικοποιείται και το εκτυπώνει όταν ζητηθεί.

- **FormulasApodotikohtas.py**

- Περιέχει τους δείκτες Αποδοτικότητας:
- Αριθμοδείκτης Μεικτού Περιθωρίου
- Αποδοτικότητα Απασχολούμενων Κεφαλαίων Τιμή Μετοχής
- P.O.A. (πρώτος τρόπος)
- P.O.E.
- Αριθμοδείκτης Οικονομικής Μόχλευσης

Το αρχείο περιέχει τον υπολογισμό των δεικτών, σύμφωνα με το μαθηματικό τους τύπο. Σε περίπτωση σφάλματος, μια εξαίρεση τύπου *EconomicsInvalidOperantError* «πετάγεται».

- **FormulasDiaforoi.py**

- Περιέχει διάφορους δείκτες:
- P.O.A. (δεύτερος τρόπος)
- P.O.I.

Το αρχείο περιέχει τον υπολογισμό των δεικτών, σύμφωνα με το μαθηματικό τους τύπο. Σε περίπτωση σφάλματος, μια εξαίρεση τύπου *EconomicsInvalidOperantError* «πετάγεται».

- **FormulasDiarthrwshsKaiViwsimothtas.py**

- Περιέχει τους δείκτες Διάρθρωσης και Βιωσιμότητας:
- Αριθμοδείκτης Ιδίων Προς Συνολικά Κεφάλαια

- Αριθμοδείκτης Ιδίων Προς Ξένα Κεφάλαια
- Αριθμοδείκτης Ιδίων Προς Πάγια Κεφάλαια
- Αριθμοδείκτης Παγίων Κεφαλαίων Προς Μακροχρόνιες Υποχρεώσεις
- Αριθμοδείκτης Κυκλοφορούντος Ενεργητικού Προς Συνολικές Υποχρεώσεις
- Αριθμοδείκτης Κάλυψης Τόκων

Το αρχείο περιέχει τον υπολογισμό των δεικτών, σύμφωνα με το μαθηματικό τους τύπο. Σε περίπτωση σφάλματος, μια εξαίρεση τύπου *EconomicsInvalidOperantError* «πετάγεται».

- **FormulasDrastiriothtas.py**

- Περιέχει τους δείκτες Δραστηριότητας:
- Αριθμοδείκτης Ταχύτητας Είσπραξης Απαιτήσεων
- Αριθμοδείκτης Ταχύτητας Βραχυπρόθεσμων Υποχρεώσεων
- Αριθμοδείκτης Ταχύτητας Κυκλοφορίας Αποθεμάτων
- Μ.Ο. Ενεργητικού
- Αριθμοδείκτης Ταχύτητας Κυκλοφορίας Καθαρού Κεφαλαίου
- Αριθμοδείκτης Ταχύτητας Κυκλοφορίας Ιδίων Κεφαλαίων
- Αριθμοδείκτης Ταχύτητας Κυκλοφορίας Παγίων Κεφαλαίων

Το αρχείο περιέχει τον υπολογισμό των δεικτών, σύμφωνα με το μαθηματικό τους τύπο. Σε περίπτωση σφάλματος, μια εξαίρεση τύπου *EconomicsInvalidOperantError* «πετάγεται».

- **FormulasEpendytikoi.py**

- Περιέχει τους Επενδυτικούς δείκτες:
- Κέρδη ανά Μετοχή
- Μέρισμα ανά Μετοχή
- Μερισματική Απόδοση
- Αριθμοδείκτης Κάλυψης Καταβαλλόμενων Μερισμάτων
- Εσωτερική Αξία Μετοχής

- Π προς Ε.Π.Σ.

Το αρχείο περιέχει τον υπολογισμό των δεικτών, σύμφωνα με το μαθηματικό τους τύπο. Σε περίπτωση σφάλματος, μια εξαίρεση τύπου *EconomicsInvalidOperantError* «πετάγεται».

- **FormulasRefstothtas.py**

- Περιέχει τους δείκτες Ρευστότητας:
- Αριθμοδείκτης Γενικής Ρευστότητας
- Αριθμοδείκτης Ειδικής Ρευστότητας
- Αριθμοδείκτης Ταμειακής Ρευστότητας
- Αριθμοδείκτης Αμυντικού Χρονικού Διαστήματος

Το αρχείο περιέχει τον υπολογισμό των δεικτών, σύμφωνα με το μαθηματικό τους τύπο. Σε περίπτωση σφάλματος, μια εξαίρεση τύπου *EconomicsInvalidOperantError* «πετάγεται».

- **__init__.py**

- Αρχείο που υπάρχει σε κάθε κατάλογο της Python 2.X για να σηματοδοτήσει ότι πρόκειται για κατάλογο Python

3.3. Templates

Τα αρχεία HTML που παρέχονται στο χρήστη και χρησιμοποιεί το σύστημα είναι τα εξής:

- **base.html**
 - Αυτό το αρχείο περιέχει τη βασική ιστοσελίδα από την οποία κληρονομούν όλες οι άλλες. Φορτώνει τα CSS/JavaScript αρχεία, τη βασική δομή της σελίδας και το μενού πλοήγησης. Ύστερα η κάθε μία από τις άλλες σελίδες εισάγει το περιεχόμενο της στη θέση που υποδεικνύεται από τα Python Tags.
- **browseInputs.html**
 - Η σελίδα που μας δείχνει το πινακάκι με τις υπάρχουσες Εισόδους Χρήστη και τις διάφορες ενέργειες που μπορούμε να κάνουμε σε αυτές (Αντιγραφή, Διαγραφή, παραγωγή Αναφοράς, κτλ).
- **browseReports.html**
 - Η σελίδα που μας δείχνει το πινακάκι με τις υπάρχουσες Αναφορές και τις διάφορες ενέργειες που μπορούμε να κάνουμε σε αυτές (Διαγραφή, θέαση Στοιχείων Εισόδου, κτλ).
- **createInput.html**
 - Η σελίδα που μας δείχνει τη φόρμα δημιουργίας Εισόδου Χρήστη
- **editInput.html**
 - Η σελίδα που μας δείχνει τη φόρμα επεξεργασίας Εισόδου Χρήστη
- **errorMessagePage.html**
 - Σελίδα που μας δείχνει μηνύματα λάθους της εφαρμογής
- **index.html**

- η κεντρική σελίδα της εφαρμογής. Μας δείχνει πόσες Είσοδοι Χρήστη και πόσες Αναφορές υπάρχουν στο σύστημα.
- **login.html**
 - το αρχείο εισόδου στο σύστημα (με όνομα χρήστη και μυστικό κωδικό)
- **logout.html**
 - Επιβεβαίωση εξόδου από το σύστημα
- **showReport.html**
 - Η σελίδα που μας δείχνει το εκτυπώσιμο πινακάκι με την Αναφορά. Η μόνη που δεν κληρονομεί από την “base.html” καθώς δε θέλουμε να εμφανίζεται μενού επάνω και φορτώνουμε άλλο CSS.

3.4. Υπόλοιπα Python αρχεία στο φάκελο *economics*

Τα υπόλοιπα Python (.py) αρχεία στο φάκελο *economics* είναι τα εξής:

- **admin.py**

Περιέχει τη διαρρύθμιση του διαχειριστικού κομματιού που παράγει αυτόματα το Django. Πιο συγκεκριμένα, παραμετροποιεί δύο διαχειριστικά panels.

1. **InputAdmin**: για την Είσοδο Χρήστη
2. **OutputAdmin**: για τις Αναφορές

- **EconomicsForms.py**

Περιέχει τη φόρμα **EconomicsInputForm** η οποία χρησιμοποιείται τόσο για τη δημιουργία Εισόδου Χρήστη, όσο και για την επεξεργασία της.

Η λογική πίσω από αυτή είναι ότι κληρονομούμε από την **ModelForm**, η οποία είναι μια αυτοματοποιημένη κλάση δημιουργίας φορμών του Django. Έχει χρησιμοποιηθεί η παραμετροποίηση:

```
exclude = ('dateSaved', 'dateCreated',)
```

για να μην εμφανίζονται τα πεδία «ημερομηνία αποθήκευσης» και «ημερομηνία δημιουργίας» στη φόρμα. Αυτά συμπληρώνονται αυτόματα από τον κώδικα της λογικής της εφαρμογής.

- **EconomicsUtils.py**

Περιέχει τις εξής βοηθητικές συναρτήσεις για την ορθή και πλήρη λειτουργία της εφαρμογής:

1. **copyEconomicsInputToOutput**: Αντιγράφει ένα αντικείμενο τύπου Είσοδος Χρήστη σε ένα αντικείμενο τύπου Αναφορά. Τα επιπλέον πεδία μένουν κενά. Αυτή η συνάρτηση χρησιμοποιείται για την αρχικοποίηση μιας Αναφοράς από το αντικείμενο που την παράγει.

2. **generateReport**: η μέθοδος αυτή παράγει την αναφορά με είσοδο ένα αντικείμενο Εισόδου Χρήστη. Για κάθε ένα δείκτη καλεί τον υπολογισμό του από τη βιβλιοθήκη και εισάγει την παραγόμενη τιμή στην αντίστοιχη μεταβλητή. Σε περίπτωση σφάλματος η εξαίρεση (exception) μεταφέρεται (propagates) στην καλούσα μέθοδο

- **models.py**

Στο αρχείο αυτό δηλώνονται τα **μοντέλα βάσης δεδομένων** της εφαρμογής μας. Αυτό το αρχείο διαβάζει το Django και δημιουργεί τα αρχεία μεταβολής βάσης στο φάκελο **migrations**.

Παρέχονται δύο μοντέλα δεδομένων (που αναλύθηκαν εκτενώς στις παραγράφους σχεδιασμού βάσης και δε θα αναλυθούν εκ νέου):

1. Είσοδος Χρήστη (**EconomicsUserInput**)
2. Αναφορά (**EconomicsOutput**)

- **urls.py**

Το Django χρησιμοποιεί το αρχείο αυτό για να μπορέσει να συνδέσει τη διεύθυνση εντοπισμού (URL) που πληκτρολογεί ο χρήστης με τον κατάλληλο κώδικα ελεγκτή (controller). Έχουν δηλωθεί οι εξής διευθύνσεις. Να σημειωθεί ότι όλες βρίσκονται κάτω από το πρόθεμα “economics”, δηλαδή ο χρήστης πρέπει να πληκτρολογήσει “όνομα_server/economics/δηλωμένη_διεύθυνση”:

- **\$** : κενό. Κεντρική σελίδα
- **createCompanyData/id** : δημιουργία δεδομένων Εισόδου Χρήστη
- **editCompanyData/id** : επεξεργασία δεδομένων Εισόδου Χρήστη
- **browseCompanyData** : συνοπτική θέαση όλων των δεδομένων Εισόδου Χρήστη

- *deleteCompanyData /id* : διαγραφή δεδομένων Εισόδου Χρήστη
- *cloneCompanyData /id* : αντιγραφή δεδομένων Εισόδου Χρήστη σε νέα οντότητα για επεξεργασία
- *createReport* : δημιουργία Αναφοράς
- *browseReports* : συνοπτική θέαση όλων των Αναφορών
- *deleteReport/id* : διαγραφή Αναφοράς
- *showReport/id* : θέαση Αναφοράς (για εκτύπωση)

- **views. py**

Στο αρχείο αυτό βρίσκεται ο κώδικας των ελεγκτών (controllers) καθώς και τυχόν βοηθητικές μέθοδοί τους. Συγκεκριμένα έχουμε τις εξής μεθόδους:

- **get_item:** βοηθητική συνάρτηση για τη γλώσσα που χρησιμοποιείται μέσα στα HTML. Επιστρέφει από ένα dictionary την τιμή που σχετίζεται με ένα key.
- **indexView:** Η κεντρική σελίδα της εφαρμογής. Επιστρέφει από τη βάση τον αριθμό των Εισόδων Χρήστη και των Αναφορών και τις περνάει στο HTML Template.
- **createCompanyDataView:** Δημιουργία Εισόδων Χρήστη. Ετοιμάζει τη φόρμα αν πρόκειται για GET μέθοδο. Αν πρόκειται για POST, διαβάζει τη φόρμα, ελέγχει αν είναι έγκυρη (valid) και αν όλα είναι OK τη σώζει στη βάση και επιστρέφει την επιβεβαίωση στο χρήστη.
- **editCompanyDataView:** Επεξεργασία Εισόδου Χρήστη. Ετοιμάζει τη φόρμα αν πρόκειται για GET μέθοδο, με βάση το id με το οποίο καλείται. Αν πρόκειται για POST, διαβάζει τη φόρμα, ελέγχει αν είναι έγκυρη (valid) και αν όλα είναι OK, σώζει τις νέες τιμές στη βάση και επιστρέφει την επιβεβαίωση στο χρήστη. Σε περίπτωση σημαντικού λάθους εμφανίζει τη σελίδα λάθους στο χρήστη.

- **browseCompanyDataView:** Συνοπτική Θέαση Εισόδων Χρήστη. Επιστρέφει από τη βάση τα δεδομένα όλων των Εισόδων Χρήστη και τις περνάει στο HTML Template.
- **deleteCompanyDataView:** Διαγραφή Εισόδου Χρήστη. Διαγράφει με βάση το id με το οποίο καλείται. Σε περίπτωση σημαντικού λάθους εμφανίζει τη σελίδα λάθους στο χρήστη. Σε επιτυχή διαγραφή επιστρέφει θετική επιβεβαίωση.
- **cloneCompanyDataView:** Κλωνοποίηση Εισόδου Χρήστη. Αντιγράφει τις τιμές όλων των μεταβλητών σε ένα νέο αντικείμενο, το οποίο και μετά σώζει στη βάση. Όλα γίνονται με βάση το id με το οποίο καλείται. Σε περίπτωση σημαντικού λάθους εμφανίζει τη σελίδα λάθους στο χρήστη.
- **createReportView:** Δημιουργία Αναφοράς. Καλεί τη βοηθητική μέθοδο και δημιουργεί μια αναφορά από την Είσοδο Χρήστη για την οποία καλείται (id) . Σε περίπτωση σημαντικού λάθους εμφανίζει τη σελίδα λάθους στο χρήστη.
- **browseReportsView:** Συνοπτική Θέαση Αναφορών. Επιστρέφει από τη βάση τα δεδομένα όλων των Αναφορών και τις περνάει στο HTML Template.
- **deleteReportView:** Διαγραφή Αναφοράς. Διαγράφει με βάση το id με το οποίο καλείται. Σε περίπτωση σημαντικού λάθους εμφανίζει τη σελίδα λάθους στο χρήστη. Σε επιτυχή διαγραφή επιστρέφει θετική επιβεβαίωση.
- **showReportView:** Θέαση Αναφοράς (για εκτύπωση). Διαβάζει από τη βάση τα στοιχεία της Αναφοράς για το id με το οποίο καλείται και τα εμφανίζει στο χρήστη. Σε περίπτωση σημαντικού λάθους εμφανίζει τη σελίδα λάθους στο χρήστη.

3.5. Υπόλοιπα Python αρχεία στο φάκελο *EconomicsWebsite*

- **wsgi.py**

Χρησιμοποιείται μόνο αν θέλουμε να χρησιμοποιήσουμε WSGI για την εκτέλεση της εφαρμογής μας. Δεν το χρησιμοποιούμε.

- **urls.py**

Το Django χρησιμοποιεί το αρχείο αυτό για να μπορέσει να συνδέσει τη διεύθυνση εντοπισμού (URL) που πληκτρολογεί ο χρήστης με τον κατάλληλο κώδικα ελεγκτή (controller). Έχουν δηλωθεί οι εξής διευθύνσεις:

1. *economics/admin* : το διαχειριστικό
2. *economics/login* : είσοδος στο σύστημα
3. *economics/logout* : έξοδος από το σύστημα
4. *economics/*: εδώ δηλώνεται ότι όποτε άλλοτε βλέπουμε το πρόθεμα “economics” θα πηγαίνουμε στο “urls.py” που βρίσκεται στο φάκελο economics για αναζήτηση ελεγκτών.

- **settings.py**

Το αρχείο με τις ρυθμίσεις του Django. Τα μόνα σημεία που χρήζουν προσοχής είναι τα εξής:

1. `INSTALLED_APPS`: έχουμε προσθέσει την εφαρμογή μας στο τέλος για να φορτώνεται από το Django.

```
INSTALLED_APPS = (  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'economics'  
)
```

2. DATABASES: έχουμε προσθέσει τη σύνδεση με την MySQL βάση.

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.mysql',  
        'NAME': 'economicsDB',  
        'USER': 'root',  
        'PASSWORD': 'user',  
        'HOST': 'localhost',  
        'PORT': '3306',  
    }  
}
```

3. TIME_ZONE: έχει ρυθμιστεί για Αθήνα.

```
TIME_ZONE = 'Europe/Athens'
```

4. STATIC_ROOT: έχει δηλωθεί η τοποθεσία των static αρχείων (JavaScript, CSS, κ.τ.λ.)

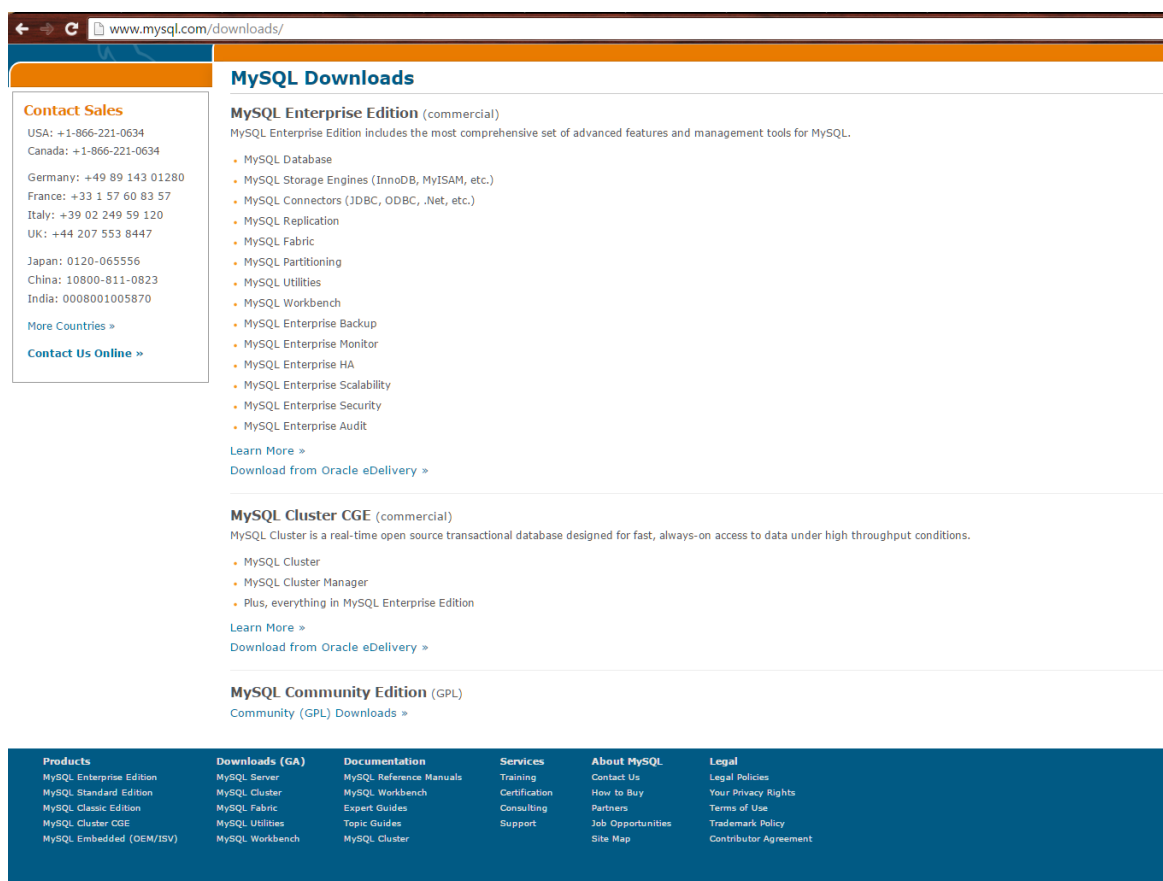
```
STATIC_ROOT = os.path.join(BASE_DIR, 'economics/static')
```


3.6. Εγκατάσταση Εφαρμογής

Παρακάτω παρατίθενται όλα τα βήματα που χρειάζονται για την επιτυχή εγκατάσταση και εκτέλεση της εφαρμογής.

3.6.1. Εγκατάσταση MySQL

Καταρχήν πρέπει να κατέβει η Community Edition της MySQL από το www.mysql.com/downloads/ και να εγκατασταθεί.



Εικόνα 8 Κατέβασμα MySQL

The screenshot shows the MySQL Community Downloads page. At the top, the MySQL logo and tagline 'The world's most popular open source database' are visible. Below this is a navigation bar with 'Downloads' highlighted. A secondary navigation bar includes 'Enterprise', 'Community', 'Yum Repository', 'APT Repository', 'SUSE Repository', 'Windows', and 'Archives'. The 'Community' section is active, showing a list of download options in the sidebar: MySQL on Windows, MySQL Yum Repository, MySQL APT Repository, MySQL SUSE Repository, MySQL Community Server, MySQL Cluster, MySQL Fabric, MySQL Utilities, MySQL Workbench, MySQL Proxy, MySQL Connectors, and Other Downloads. The main content area displays three download options:

- MySQL Community Server (GPL)**: (Current Generally Available Release: 5.6.26). MySQL Community Server is the world's most popular open source database. [DOWNLOAD](#)
- MySQL Cluster (GPL)**: (Current Generally Available Release: 7.4.7). MySQL Cluster is a real-time, open source transactional database. [DOWNLOAD](#)
- MySQL Fabric (GPL)**: MySQL Fabric provides a framework for managing High Availability and Sharding. [DOWNLOAD](#)

Εικόνα 9 MySQL Community Server Download - Βήμα 2

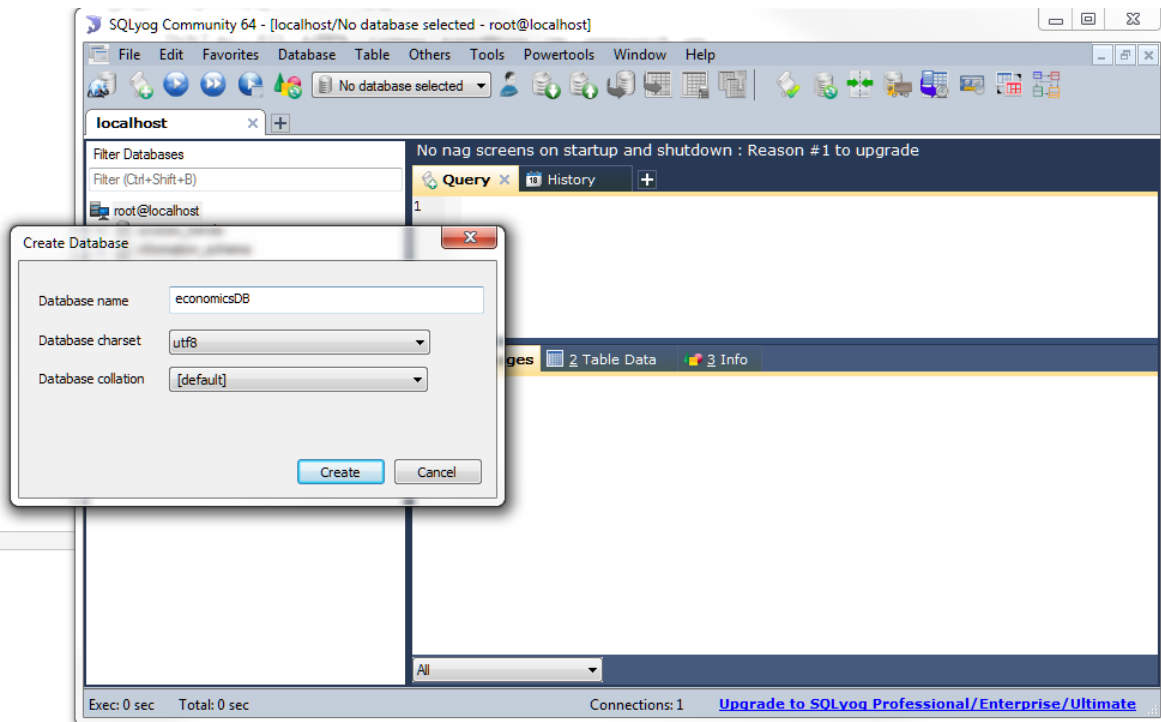
The screenshot shows the MySQL Community Server 5.6.26 download page. It features two tabs: 'Generally Available (GA) Releases' (selected) and 'Development Releases'. The page title is 'MySQL Community Server 5.6.26'. A 'Select Platform:' dropdown menu is set to 'Microsoft Windows'. Under 'Recommended Download:', there is a large graphic for 'MySQL Installer 5.6 for Windows' with the text 'All MySQL Products. For All Windows Platforms. In One Package.' and a 'Download' button. Below this, under 'Other Downloads:', there is a table listing additional download options:

| Platform | Version | Size | Action |
|--|---------|--------|--------------------------|
| Windows (x86, 32-bit), ZIP Archive | 5.6.26 | 343.9M | Download |
| <small>(mysql-5.6.26-win32.zip) MD5: 6a96de88e2aaF872a228fca7da85dfed Signature</small> | | | |
| Windows (x86, 64-bit), ZIP Archive | 5.6.26 | 349.4M | Download |
| <small>(mysql-5.6.26-winx64.zip) MD5: 4fe28f844dd39e9f41549a3792642b07 Signature</small> | | | |

At the bottom, a warning icon and text state: 'We suggest that you use the MD5 checksums and GnuPG signatures to verify the integrity of the packages you download.'

Εικόνα 10 MySQL Installer - Εγκατάσταση MySQL βήμα 3

Μετά την εγκατάσταση της MySQL πρέπει να φτιάξουμε μια βάση με όνομα **economicsDB** σε UTF8 κωδικοποίηση. Στην Εικόνα 11 αυτό φαίνεται καθαρά. Στη συγκεκριμένη εικόνα χρησιμοποιείται το εργαλείο SQLyog (<https://www.webyog.com/>) αλλά μπορεί να γίνει και με απλή SQL (create database).

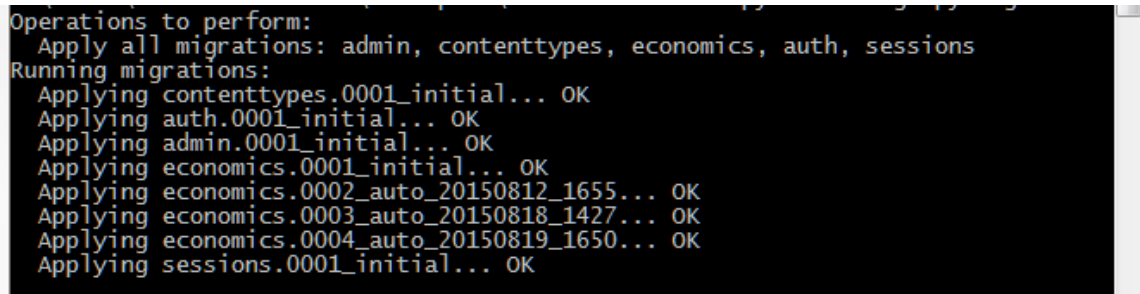


Εικόνα 11 Δημιουργία Βάσης

3.6.2. Εγκατάσταση και αρχικοποίηση εφαρμογής

Αμέσως μετά, πρέπει να γίνει το στήσιμο της βάσης. Για να γίνει αυτό, πρέπει να εκτελεστεί η εξής εντολή μέσα από τον κατάλογο της εφαρμογής:

```
python manage.py migrate
```

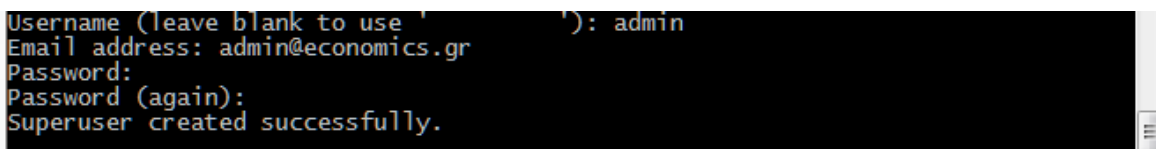


```
Operations to perform:
  Apply all migrations: admin, contenttypes, economics, auth, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying economics.0001_initial... OK
  Applying economics.0002_auto_20150812_1655... OK
  Applying economics.0003_auto_20150818_1427... OK
  Applying economics.0004_auto_20150819_1650... OK
  Applying sessions.0001_initial... OK
```

Εικόνα 12 Στήσιμο βάσης

Και να φτιάξουμε έναν αρχικό διαχειριστή/χρήστη της εφαρμογής. Με την εντολή **python manage.py createsuperuser**

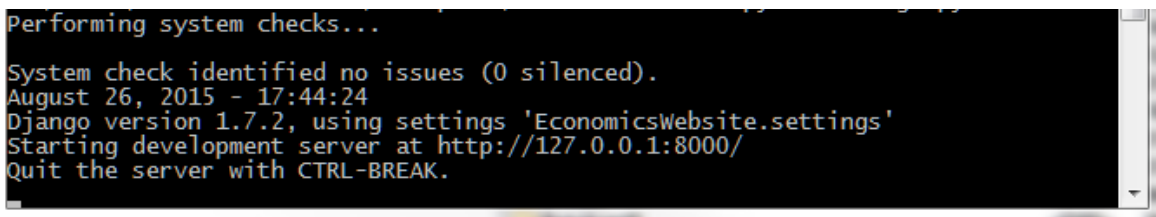
Παρακάτω φτιάχνεται ένας με όνομα admin και κωδικό admin.



```
Username (leave blank to use 'admin'): admin
Email address: admin@economics.gr
Password:
Password (again):
Superuser created successfully.
```

Εικόνα 13 Δημιουργία αρχικού χρήστη

Τέλος, για να εκκινήσει ο server, πρέπει να εκτελεστεί η εξής εντολή μέσα από τον κατάλογο της εφαρμογής: **python manage.py runserver**



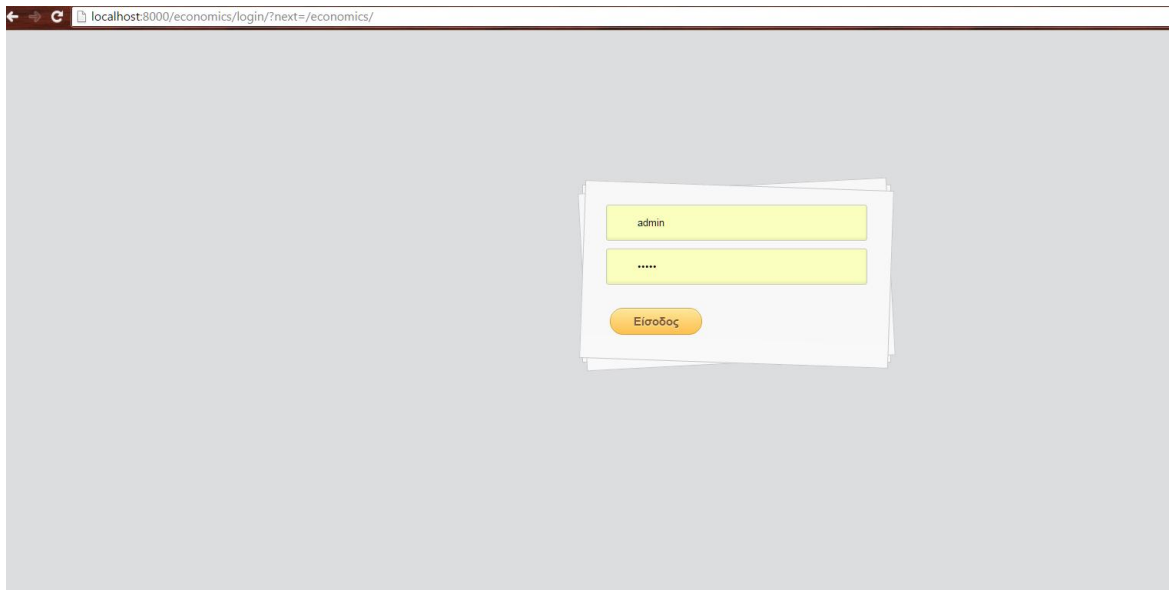
```
Performing system checks...

System check identified no issues (0 silenced).
August 26, 2015 - 17:44:24
Django version 1.7.2, using settings 'EconomicsWebsite.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Εικόνα 14 Εκκίνηση εξυπηρετητή

Πλέον μπορούμε να δούμε την εφαρμογή στο:

<http://localhost:8000/economics/>



Εικόνα 15 αρχικό όνομα χρήστη: admin/admin

4. Μελλοντική εργασία και επεκτάσεις

Στην πτυχιακή αυτή περιγράφηκε ο τρόπος κατασκευής μιας εφαρμογής διαχείρισης Οικονομικών Στοιχείων και Δημιουργίας Αναφορών για αυτά. Αναφέρθηκαν όλα τα βήματα, από την επιλογή τεχνολογιών, μέχρι το σχεδιασμό και την τελική υλοποίηση. Η τελική εφαρμογή είναι διαλειτουργική, καθώς τρέχει σε πολλές πλατφόρμες, με ελάχιστο κόπο και πολύ ευέλικτη, κάτι που μας το επέτρεψε η χρήση της γλώσσας Python.

Υπάρχουν πολλά ακόμη που μπορεί να γίνουν σε μελλοντικές εκδόσεις της εφαρμογής:

1. Ακόμα πιο δυναμική είσοδος δεδομένων από τους χρήστες.
2. Είσοδος δεδομένων με πιο φυσικό τρόπο, ίσως με χρήση wizards (βήμα προς βήμα οδηγών)
3. Θέαση Αναφορών με πολλαπλούς τρόπους (με χρήση JavaScript)
4. Χρήση γραφημάτων για καλύτερη κατανόηση των στοιχείων των Αναφορών

Ευελπιστούμε ότι σε επόμενη έκδοση της εφαρμογής θα έχουν υλοποιηθεί τα περισσότερα από αυτά και θα υπάρχουν νέες λίστες με βελτιώσεις και επεκτάσεις.

Παράρτημα Α: Τελικός σχεδιασμός μοντέλων Django

Ακολουθεί ο κώδικας των μοντέλων του τελικού σχεδιασμού σε Django!

```
'''
Ορίζει τα χαρακτηριστικά που πρέπει να εισάγει ο χρήστης για να έχουμε
έξοδο προγράμματος
'''

class EconomicsUserInput(models.Model):
    def __unicode__(self): # Python 3: def __str__(self):
        return str(self.companyName) + " @ " + strftime(self.dateSaved, "%a, %d
        %b %Y %H:%M:%S %z")

    #Γενικά
    companyName = models.CharField(max_length=255, verbose_name=u'Όνομα
Εταιρίας')
    dateCreated = models.DateTimeField(verbose_name=u'Ημερομηνία Δημιουργίας')
    dateSaved = models.DateTimeField(verbose_name=u'Ημερομηνία Τελευταίας
Αλλαγής')

    #Αποδοτικότητα
    miktaKerdiEkmetallefsis = models.DecimalField(decimal_places=2,
    max_digits=12, verbose_name=u'Μικτά Κέρδη Εκμετάλλευσης', null=True)
    katharesPwlhseisXrisewn = models.DecimalField(decimal_places=2,
    max_digits=12, verbose_name=u'Καθαρές Πωλήσεις Χρήσεων', null=True)
    katharaKerdiEkmetallefsis = models.DecimalField(decimal_places=2,
    max_digits=12, verbose_name=u'Καθαρά Κέρδη Εκμετάλλευσης', null=True)
    xrimatooikonomikaEksoda = models.DecimalField(decimal_places=2,
    max_digits=12, verbose_name=u'Χρηματοοικονομικά Έξοδα', null=True)
    synolikaApasxoloumenaKefalaia = models.DecimalField(decimal_places=2,
    max_digits=12, verbose_name=u'Συνολικά Απασχολούμενα Κεφάλαια', null=True)
    synolikoEnerghtiko = models.DecimalField(decimal_places=2, max_digits=12,
    verbose_name=u'Συνολικό Ενεργητικό', null=True)
    synoloIdiwnKefalaiwn = models.DecimalField(decimal_places=2, max_digits=12,
    verbose_name=u'Συνολικό Ιδίων Κεφαλαίων', null=True)
    apodotikohtaIdiwnKefalaiwn = models.DecimalField(decimal_places=2,
    max_digits=12, verbose_name=u'Αποδοτικότητα Ιδίων Κεφαλαίων', null=True)
    apodotikohtaSynolouApasxoloumenwnKefalaiwn =
    models.DecimalField(decimal_places=2, max_digits=12,
    verbose_name=u'Αποδοτικότητα Συνόλου Απασχολούμενων Κεφαλαίων', null=True)

    #Διάφοροι
    katharaKerdhProForwn = models.DecimalField(decimal_places=2, max_digits=12,
    verbose_name=u'Καθαρά Κέρδη Προ Φ'όρων', null=True)
    moEnerghtikou = models.DecimalField(decimal_places=2, max_digits=12,
    verbose_name=u'Μ.Ο. Ενεργητικού', null=True)
    idiaKefalaia = models.DecimalField(decimal_places=2, max_digits=12,
    verbose_name=u'Ιδία Κεφάλαια', null=True)
```

```
kseNaKefalaia = models.DecimalField(decimal_places=2, max_digits=12, verbose_name=u'Ξένα Κεφάλαια', null=True)
```

#Διάρθρωση Και Βιωσιμότητα

```
synolikaKefalaia = models.DecimalField(decimal_places=2, max_digits=12, verbose_name=u'Συνολικά Κεφάλαια', null=True)  
pagiaKefalaia = models.DecimalField(decimal_places=2, max_digits=12, verbose_name=u'Πάγια Κεφάλαια', null=True)  
makroxroniesYpoxrewseis = models.DecimalField(decimal_places=2, max_digits=12, verbose_name=u'Μακροχρόνιες Υποχρεώσεις', null=True)  
kykloforounEnerghtiko = models.DecimalField(decimal_places=2, max_digits=12, verbose_name=u'Κυκλοφορούν Ενεργητικό', null=True)  
synoloYpoxrewsewn = models.DecimalField(decimal_places=2, max_digits=12, verbose_name=u'Σύνολο Υποχρεώσεων', null=True)  
synoloTokwn = models.DecimalField(decimal_places=2, max_digits=12, verbose_name=u'Σύνολο Τόκων', null=True)
```

#Δραστηριότητα

```
katharesPwlhseis = models.DecimalField(decimal_places=2, max_digits=12, verbose_name=u'Καθαρές Πωλήσεις', null=True)  
mesosOrosApaithsewn = models.DecimalField(decimal_places=2, max_digits=12, verbose_name=u'Μέσος Όρος Απαιτήσεων', null=True)  
kostosPwlithentwn = models.DecimalField(decimal_places=2, max_digits=12, verbose_name=u'Κόστος Πωληθέντων', null=True)  
moby = models.DecimalField(decimal_places=2, max_digits=12, verbose_name=u'M.O.B.Y', null=True)  
mesoApothemaProiontwn = models.DecimalField(decimal_places=2, max_digits=12, verbose_name=u'Μέσο Απόθεμα Προϊόντων', null=True)  
energhtikoArxhs = models.DecimalField(decimal_places=2, max_digits=12, verbose_name=u'Ενεργητικό Αρχής', null=True)  
energhtikoTelous = models.DecimalField(decimal_places=2, max_digits=12, verbose_name=u'Ενεργητικό Τέλους', null=True)  
pagioEnerghtiko = models.DecimalField(decimal_places=2, max_digits=12, verbose_name=u'Πάγιο Ενεργητικό', null=True)
```

#Επενδυτικοί

```
synoloKatharwnKerdwN = models.DecimalField(decimal_places=2, max_digits=12, verbose_name=u'Σύνολο Καθαρών Κερδών', null=True)  
arithmosMetoxwn = models.DecimalField(decimal_places=2, max_digits=12, verbose_name=u'Αριθμός Μετοχών', null=True)  
synoloMerismatwn = models.DecimalField(decimal_places=2, max_digits=12, verbose_name=u'Σύνολο Μερισμάτων', null=True)  
arithmosMetoxwnSeKykloforia = models.DecimalField(decimal_places=2, max_digits=12, verbose_name=u'Αριθμός Μετοχών σε Κυκλοφορία', null=True)  
merismaAnaMetoxh = models.DecimalField(decimal_places=2, max_digits=12, verbose_name=u'Μέρισμα Ανά Μετοχή', null=True)  
timhMetoxhs = models.DecimalField(decimal_places=2, max_digits=12, verbose_name=u'Τιμή Μετοχής', null=True)  
katharaKerdhXrhsewn = models.DecimalField(decimal_places=2, max_digits=12, verbose_name=u'Καθαρά Κέρδη Χρήσεων', null=True)  
synoloKatavallomenwnMerismatwn = models.DecimalField(decimal_places=2, max_digits=12, verbose_name=u'Σύνολο Καταβαλλόμενων Μερισμάτων', null=True)
```



```

xrhmatisthriakhTimh = models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Χρηματιστηριακή Τιμή', null=True)
kerdhAnaMetoxh = models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Κέρδη Ανά Μετοχή', null=True)

#Ρευστότητα
diathesima = models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Διαθέσιμα', null=True)
apaithseis = models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Απαιτήσεις', null=True)
apothemata = models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Αποθέματα', null=True)
vraxyprothesmesYpoxrewseis = models.DecimalField(decimal_places=2,
max_digits=12, verbose_name=u'Βραχυπρόθεσμες Υποχρεώσεις', null=True)
diathesimoEnerghtiko = models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Διαθέσιμο Ενεργητικό', null=True)
lhksiprothesmesYpoxrewseis = models.DecimalField(decimal_places=2,
max_digits=12, verbose_name=u'Ληξιπρόθεσμες Υποχρεώσεις', null=True)
provlepomenesHmerisiesLeitourgikesDapanes =
models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Προβλεπόμενες Ημερήσιες Λειτουργικές Δαπάνες', null=True)

class Meta:
    unique_together = ("companyName", "dateSaved") # δεν μπορούν να
υπάρχουν δύο είσοδοι χρήστη με το ίδιο όνομα εταιρίας, την ίδια στιγμή

'''
Ορίζει την έξοδο προγράμματος με δεδομένη είσοδο από το χρήστη
'''

class EconomicsOutput(models.Model):
    def __unicode__(self): # Python 3: def __str__(self):
        return str(self.companyName) + " @ " + strftime(self.dateGenerated,
"%a, %d %b %Y %H:%M:%S %z")

    #Γενικά
    dateGenerated = models.DateTimeField(verbose_name=u'Ημερομηνία
Αποθήκευσης')

    # OUTPUT
    #Αποδοτικότητα
    arithmodeiktisMeiktouPerithoriou = models.DecimalField(decimal_places=2,
max_digits=12, verbose_name=u'Αριθμοδείκτης Μεικτού Περιθωρίου', null=True)
apodotikotitaApasxoloumenwnKefalaiwn =
models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Αποδοτικότητα Απασχολούμενων Κεφαλαίων', null=True)
roa = models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'P.O.A.', null=True)
roe = models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'P.O.E.', null=True)
arithmodeiktisOikonomikisMoxlefsis = models.DecimalField(decimal_places=2,
max_digits=12, verbose_name=u'Αριθμοδείκτης Οικονομικής Μόχλευσης', null=True)

```

```

#Διάφοροι
roa_B = models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Ρ.Ο.Α. (Β τρόπος)', null=True)
roi = models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Ρ.Ο.Ι.', null=True)

#Διάρθρωση Και Βιωσιμότητα
arithmodeiktisIdiaProsSynolikaKefalaia =
models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Αριθμοδείκτης Ιδίων Προς Συνολικά Κεφάλαια', null=True)
arithmodeiktisIdiaProsKsenaKefalaia = models.DecimalField(decimal_places=2,
max_digits=12, verbose_name=u'Αριθμοδείκτης Ιδίων Προς Ξένα Κεφάλαια',
null=True)
arithmodeiktisIdiaProsPagiaKefalaia = models.DecimalField(decimal_places=2,
max_digits=12, verbose_name=u'Αριθμοδείκτης Ιδίων Προς Πάγια Κεφάλαια',
null=True)
arithmodeiktisPagiaKefalaiaProsMakroxroniesYpoxrewseis =
models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Αριθμοδείκτης Παγίων Κεφαλαίων Προς Μακροχρονίες Υποχρεώσεις',
null=True)
arithmodeiktisKykloforountosEnerghtikouProsSynolikesYpoxrewseis =
models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Αριθμοδείκτης Κυκλοφορούντος Ενεργητικού Προς Συνολικές
Υποχρεώσεις', null=True)
arithmodeiktisKalypshsTokwn = models.DecimalField(decimal_places=2,
max_digits=12, verbose_name=u'Αριθμοδείκτης Κάλυψης Τόκων', null=True)

#Δραστηριότητα
arithmodeiktisTaxythtasEisprakshsAraithsewn =
models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Αριθμοδείκτης Ταχύτητας Είσπραξης Απαιτήσεων', null=True)
arithmodeiktisTaxythtasVraxyprothesmwnYpoxrewsewn =
models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Αριθμοδείκτης Ταχύτητας Βραχυπρόθεσμων Υποχρεώσεων', null=True)
arithmodeiktisTaxythtasKykloforiasApothematwn =
models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Αριθμοδείκτης Ταχύτητας Κυκλοφορίας Αποθεμάτων', null=True)
mesoOroEnerghtikou = models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Μ.Ο. Ενεργητικού', null=True)
arithmodeiktisTaxythtasKykloforiasKatharouKefalaiou =
models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Αριθμοδείκτης Ταχύτητας Κυκλοφορίας Καθαρού Κεφαλαίου',
null=True)
arithmodeiktisTaxythtasKykloforiasIdiwnKefalaiwn =
models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Αριθμοδείκτης Ταχύτητας Κυκλοφορίας Ιδίων Κεφαλαίων', null=True)
arithmodeiktisTaxythtasKykloforiasPagiwn =
models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Αριθμοδείκτης Ταχύτητας Κυκλοφορίας Παγίων Κεφαλαίων',
null=True)

#Επενδυτικοί

```

```

kerdhAnaMetoxh = models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Κέρδη ανά Μετοχή', null=True)
merismaAnaMetoxh = models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Μέρισμα ανά Μετοχή', null=True)
merismatikhApodosh = models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Μερισματική Απόδοση', null=True)
arithmodeikthsKalypshsKatavallomenwnMerismatwn =
models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Αριθμοδείκτης Κάλυψης Καταβαλλόμενων Μερισμάτων', null=True)
eswterikhAksiaMetoxhs = models.DecimalField(decimal_places=2,
max_digits=12, verbose_name=u'Εσωτερική Αξία Μετοχής', null=True)
pProsEPS = models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Π προς Ε.Π.Σ.', null=True)

#Ρευστότητα
arithmodeiktisGenikisRefstothtas = models.DecimalField(decimal_places=2,
max_digits=12, verbose_name=u'Αριθμοδείκτης Γενικής Ρευστότητας', null=True)
arithmodeiktisEidikisRefstothtas = models.DecimalField(decimal_places=2,
max_digits=12, verbose_name=u'Αριθμοδείκτης Ειδικής Ρευστότητας', null=True)
arithmodeiktisTameiakisRefstothtas = models.DecimalField(decimal_places=2,
max_digits=12, verbose_name=u'Αριθμοδείκτης Ταμειακής Ρευστότητας', null=True)
arithmodeiktisAmyntikouXronikouDiasthmatos =
models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Αριθμοδείκτης Αμυντικού Χρονικού Διαστήματος', null=True)

# INPUT
#Γενικά
companyName = models.CharField(max_length=255, verbose_name=u'Όνομα
Εταιρίας')
dateCreated = models.DateTimeField(verbose_name=u'Ημερομηνία Δημιουργίας')
dateSaved = models.DateTimeField(verbose_name=u'Ημερομηνία Τελευταίας
Αλλαγής')

#Αποδοτικότητα
miktaKerdiEkmatallefsis = models.DecimalField(decimal_places=2,
max_digits=12, verbose_name=u'Μικτά Κέρδη Εκμετάλλευσης', null=True)
katharesPwlhseisXrisewn = models.DecimalField(decimal_places=2,
max_digits=12, verbose_name=u'Καθαρές Πωλήσεις Χρήσεων', null=True)
katharaKerdiEkmatallefsis = models.DecimalField(decimal_places=2,
max_digits=12, verbose_name=u'Καθαρά Κέρδη Εκμετάλλευσης', null=True)
xrimatooikonomikaEksoda = models.DecimalField(decimal_places=2,
max_digits=12, verbose_name=u'Χρηματοοικονομικά Έξοδα', null=True)
synolikaApasxoloumenaKefalaia = models.DecimalField(decimal_places=2,
max_digits=12, verbose_name=u'Συνολικά Απασχολούμενα Κεφάλαια', null=True)
synolikoEnerghtiko = models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Συνολικό Ενεργητικό', null=True)
synoloIdiwnKefalaiwn = models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Συνολικό Ιδίων Κεφαλαίων', null=True)
apodotikothtaIdiwnKefalaiwn = models.DecimalField(decimal_places=2,
max_digits=12, verbose_name=u'Αποδοτικότητα Ιδίων Κεφαλαίων', null=True)
apodotikothtaSynolouApasxoloumenwnKefalaiwn =
models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Αποδοτικότητα Συνόλου Απασχολούμενων Κεφαλαίων', null=True)

```

```

#Διάφοροι
katharaKerdhProForwn = models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Καθαρά Κέρδη Προ Φόρων', null=True)
moEnerghtikou = models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'M.O. Ενεργητικού', null=True)
idiaKefalaia = models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Ιδία Κεφάλαια', null=True)
ksenaKefalaia = models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Ξένα Κεφάλαια', null=True)

#Διάρθρωση Και Βιωσιμότητα
synolikaKefalaia = models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Συνολικά Κεφάλαια', null=True)
pagiaKefalaia = models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Πάγια Κεφάλαια', null=True)
makroxroniesYpoxrewseis = models.DecimalField(decimal_places=2,
max_digits=12, verbose_name=u'Μακροχρόνιες Υποχρεώσεις', null=True)
kykloforounEnerghtiko = models.DecimalField(decimal_places=2,
max_digits=12, verbose_name=u'Κυκλοφορούν Ενεργητικό', null=True)
synoloYpoxrewsewn = models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Σύνολο Υποχρεώσεων', null=True)
synoloTokwn = models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Σύνολο Τόκων', null=True)

#Δραστηριότητα
katharesPwlhseis = models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Καθαρές Πωλήσεις', null=True)
mesosOrosApaithsewn = models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Μέσος Όρος Απαιτήσεων', null=True)
kostosPwlithentwn = models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Κόστος Πωληθέντων', null=True)
moby = models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'M.O.B.Y', null=True)
mesoApothemaProiontwn = models.DecimalField(decimal_places=2,
max_digits=12, verbose_name=u'Μέσο Απόθεμα Προϊόντων', null=True)
energhtikoArxhs = models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Ενεργητικό Αρχής', null=True)
energhtikoTelous = models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Ενεργητικό Τέλους', null=True)
pagioEnerghtiko = models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Πάγιο Ενεργητικό', null=True)

#Επενδυτικοί
synoloKatharwnKerdwn = models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Σύνολο Καθαρών Κερδών', null=True)
arithmosMetoxwn = models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Αριθμός Μετοχών', null=True)
synoloMerismatwn = models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Σύνολο Μερισμάτων', null=True)
arithmosMetoxwnSeKykloforia = models.DecimalField(decimal_places=2,
max_digits=12, verbose_name=u'Αριθμός Μετοχών σε Κυκλοφορία', null=True)

```

```

merismaAnaMetoxh = models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Μέρισμα Ανά Μετοχή', null=True)
timhMetoxhs = models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Τιμή Μετοχής', null=True)
katharaKerdhXrhsewn = models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Καθαρά Κέρδη Χρήσεων', null=True)
synoloKatavallomenwnMerismatwn = models.DecimalField(decimal_places=2,
max_digits=12, verbose_name=u'Σύνολο Καταβαλλόμενων Μερισμάτων', null=True)
xrhmatisthriakhTimh = models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Χρηματιστηριακή Τιμή', null=True)
kerdhAnaMetoxh = models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Κέρδη Ανά Μετοχή', null=True)

#Ρευστότητα
diathesima = models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Διαθέσιμα', null=True)
apaitiseis = models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Απαιτήσεις', null=True)
apothemata = models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Αποθέματα', null=True)
vraxyprothesmesYpoxrewseis = models.DecimalField(decimal_places=2,
max_digits=12, verbose_name=u'Βραχυπρόθεσμες Υποχρεώσεις', null=True)
diathesimoEnerghtiko = models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Διαθέσιμο Ενεργητικό', null=True)
lhksiprothesmesYpoxrewseis = models.DecimalField(decimal_places=2,
max_digits=12, verbose_name=u'Ληξιπρόθεσμες Υποχρεώσεις', null=True)
provlepomenesHmerisiesLeitourgikesDapanes =
models.DecimalField(decimal_places=2, max_digits=12,
verbose_name=u'Προβλεπόμενες Ημερήσιες Λειτουργικές Δαπάνες', null=True)

```

Παράρτημα Β: SQL-βάση μοντέλων Django

Ακολουθεί μέρος του κώδικα SQL που παρήχθη από τα μοντέλα του τελικού σχεδιασμού σε Django. Περιέχει τις δημιουργίες πινάκων, ευρετηρίων και ξένων κλειδιών.

```
CREATE DATABASE /*!32312 IF NOT EXISTS*/`economicsdb` /*!40100 DEFAULT CHARACTER SET utf8 */;

USE `economicsdb`;

/*Table structure for table `auth_group` */

DROP TABLE IF EXISTS `auth_group`;

CREATE TABLE `auth_group` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(80) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `name` (`name`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

/*Table structure for table `auth_group_permissions` */

DROP TABLE IF EXISTS `auth_group_permissions`;

CREATE TABLE `auth_group_permissions` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `group_id` int(11) NOT NULL,
  `permission_id` int(11) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `group_id` (`group_id`,`permission_id`),
  KEY `auth_group_permissions_0e939a4f` (`group_id`),
  KEY `auth_group_permissions_8373b171` (`permission_id`),
  CONSTRAINT `auth_group_permissi_permission_id_23962d04_fk_auth_permission_id` FOREIGN KEY (`permission_id`) REFERENCES `auth_permission` (`id`),
  CONSTRAINT `auth_group_permissions_group_id_58c48ba9_fk_auth_group_id` FOREIGN KEY (`group_id`) REFERENCES `auth_group` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

/*Table structure for table `auth_permission` */

DROP TABLE IF EXISTS `auth_permission`;

CREATE TABLE `auth_permission` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(50) NOT NULL,
  `content_type_id` int(11) NOT NULL,
  `codename` varchar(100) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `content_type_id` (`content_type_id`,`codename`),
  KEY `auth_permission_417f1b1c` (`content_type_id`),
  CONSTRAINT `auth_permissi_content_type_id_51277a81_fk_django_content_type_id` FOREIGN KEY (`content_type_id`) REFERENCES `django_content_type` (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=25 DEFAULT CHARSET=utf8;
```

```

/*Table structure for table `auth_user` */

DROP TABLE IF EXISTS `auth_user`;

CREATE TABLE `auth_user` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `password` varchar(128) NOT NULL,
  `last_login` datetime NOT NULL,
  `is_superuser` tinyint(1) NOT NULL,
  `username` varchar(30) NOT NULL,
  `first_name` varchar(30) NOT NULL,
  `last_name` varchar(30) NOT NULL,
  `email` varchar(75) NOT NULL,
  `is_staff` tinyint(1) NOT NULL,
  `is_active` tinyint(1) NOT NULL,
  `date_joined` datetime NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `username` (`username`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

/*Table structure for table `auth_user_groups` */

DROP TABLE IF EXISTS `auth_user_groups`;

CREATE TABLE `auth_user_groups` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `user_id` int(11) NOT NULL,
  `group_id` int(11) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `user_id` (`user_id`,`group_id`),
  KEY `auth_user_groups_e8701ad4` (`user_id`),
  KEY `auth_user_groups_0e939a4f` (`group_id`),
  CONSTRAINT `auth_user_groups_group_id_30a071c9_fk_auth_group_id` FOREIGN KEY
(`group_id`) REFERENCES `auth_group` (`id`),
  CONSTRAINT `auth_user_groups_user_id_24702650_fk_auth_user_id` FOREIGN KEY
(`user_id`) REFERENCES `auth_user` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

/*Table structure for table `auth_user_user_permissions` */

DROP TABLE IF EXISTS `auth_user_user_permissions`;

CREATE TABLE `auth_user_user_permissions` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `user_id` int(11) NOT NULL,
  `permission_id` int(11) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `user_id` (`user_id`,`permission_id`),
  KEY `auth_user_user_permissions_e8701ad4` (`user_id`),
  KEY `auth_user_user_permissions_8373b171` (`permission_id`),
  CONSTRAINT `auth_user_user_perm_permission_id_3d7071f0_fk_auth_permission_id`
FOREIGN KEY (`permission_id`) REFERENCES `auth_permission` (`id`),
  CONSTRAINT `auth_user_user_permissions_user_id_7cd7acb6_fk_auth_user_id`
FOREIGN KEY (`user_id`) REFERENCES `auth_user` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

/*Table structure for table `django_admin_log` */

DROP TABLE IF EXISTS `django_admin_log`;

```

```

CREATE TABLE `django_admin_log` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `action_time` datetime NOT NULL,
  `object_id` longtext,
  `object_repr` varchar(200) NOT NULL,
  `action_flag` smallint(5) unsigned NOT NULL,
  `change_message` longtext NOT NULL,
  `content_type_id` int(11) DEFAULT NULL,
  `user_id` int(11) NOT NULL,
  PRIMARY KEY (`id`),
  KEY `django_admin_log_417f1b1c` (`content_type_id`),
  KEY `django_admin_log_e8701ad4` (`user_id`),
  CONSTRAINT `django_admin_log_user_id_1c5f563_fk_auth_user_id` FOREIGN KEY
(`user_id`) REFERENCES `auth_user` (`id`),
  CONSTRAINT `django_admin__content_type_id_5151027a_fk_django_content_type_id`
FOREIGN KEY (`content_type_id`) REFERENCES `django_content_type` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

/*Table structure for table `django_content_type` */

DROP TABLE IF EXISTS `django_content_type`;

CREATE TABLE `django_content_type` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(100) NOT NULL,
  `app_label` varchar(100) NOT NULL,
  `model` varchar(100) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `django_content_type_app_label_3ec8c61c_uniq`
(`app_label`,`model`)
) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=utf8;

/*Table structure for table `django_migrations` */

DROP TABLE IF EXISTS `django_migrations`;

CREATE TABLE `django_migrations` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `app` varchar(255) NOT NULL,
  `name` varchar(255) NOT NULL,
  `applied` datetime NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=utf8;

/*Table structure for table `django_session` */

DROP TABLE IF EXISTS `django_session`;

CREATE TABLE `django_session` (
  `session_key` varchar(40) NOT NULL,
  `session_data` longtext NOT NULL,
  `expire_date` datetime NOT NULL,
  PRIMARY KEY (`session_key`),
  KEY `django_session_de54fa62` (`expire_date`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

/*Table structure for table `economics_economicsoutput` */

DROP TABLE IF EXISTS `economics_economicsoutput`;

CREATE TABLE `economics_economicsoutput` (

```



```

`id` int(11) NOT NULL AUTO_INCREMENT,
`dateGenerated` datetime NOT NULL,
`arithmodeiktisMeiktouPerithoriou` decimal(12,2) DEFAULT NULL,
`apodotikotitaApasxoloumenwnKefalain` decimal(12,2) DEFAULT NULL,
`roa` decimal(12,2) DEFAULT NULL,
`roe` decimal(12,2) DEFAULT NULL,
`arithmodeiktisOikonomikisMoxlefsis` decimal(12,2) DEFAULT NULL,
`roa_B` decimal(12,2) DEFAULT NULL,
`roi` decimal(12,2) DEFAULT NULL,
`arithmodeiktisIdiaProsSynolikaKefalaia` decimal(12,2) DEFAULT NULL,
`arithmodeiktisIdiaProsKsenaKefalaia` decimal(12,2) DEFAULT NULL,
`arithmodeiktisIdiaProsPagiaKefalaia` decimal(12,2) DEFAULT NULL,
`arithmodeiktisPagiaKefalaiaProsMakroxroniesYpoxrewseis` decimal(12,2)
DEFAULT NULL,
`arithmodeiktisKykloforontosEnerghtikouProsSynolikesYpoxrewseis`
decimal(12,2) DEFAULT NULL,
`arithmodeiktisKalypshsTokwn` decimal(12,2) DEFAULT NULL,
`arithmodeiktisTaxythtasEisprakshsApaithsewn` decimal(12,2) DEFAULT NULL,
`arithmodeiktisTaxythtasVraxyprothesmwnYpoxrewsewn` decimal(12,2) DEFAULT
NULL,
`arithmodeiktisTaxythtasKykloforiasApothematwn` decimal(12,2) DEFAULT NULL,
`mesoOroEnerghtikou` decimal(12,2) DEFAULT NULL,
`arithmodeiktisTaxythtasKykloforiasKatharouKefalaiou` decimal(12,2) DEFAULT
NULL,
`arithmodeiktisTaxythtasKykloforiasIdiwnKefalain` decimal(12,2) DEFAULT
NULL,
`arithmodeiktisTaxythtasKykloforiasPagiwn` decimal(12,2) DEFAULT NULL,
`kerdhAnaMetoxh` decimal(12,2) DEFAULT NULL,
`merismaAnaMetoxh` decimal(12,2) DEFAULT NULL,
`merismatikhApodosh` decimal(12,2) DEFAULT NULL,
`arithmodeikthsKalypshsKatavallomenwnMerismatwn` decimal(12,2) DEFAULT NULL,
`eswterikhAksiaMetoxhs` decimal(12,2) DEFAULT NULL,
`pProseEPS` decimal(12,2) DEFAULT NULL,
`arithmodeiktisGenikisRefstothtas` decimal(12,2) DEFAULT NULL,
`arithmodeiktisEidikisRefstothtas` decimal(12,2) DEFAULT NULL,
`arithmodeiktisTameiakisRefstothtas` decimal(12,2) DEFAULT NULL,
`arithmodeiktisAmyntikouXronikouDiasthmatos` decimal(12,2) DEFAULT NULL,
`mesoOrosApaithsewn` decimal(12,2) DEFAULT NULL,
`apaithseis` decimal(12,2),
`apodotikohttaIdiwnKefalain` decimal(12,2),
`apodotikohttaSynolouApasxoloumenwnKefalain` decimal(12,2),
`apothemata` decimal(12,2),
`arithmosMetoxwn` decimal(12,2),
`arithmosMetoxwnSeKykloforia` decimal(12,2),
`companyName` varchar(255) NOT NULL,
`dateCreated` datetime NOT NULL,
`dateSaved` datetime NOT NULL,
`diathesima` decimal(12,2),
`diathesimoEnerghtiko` decimal(12,2),
`energhtikoArxhs` decimal(12,2),
`energhtikoTelous` decimal(12,2),
`idiaKefalaia` decimal(12,2),
`katharaKerdhProForwn` decimal(12,2),
`katharaKerdhXrhsewn` decimal(12,2),
`katharaKerdiEkmetallefsis` decimal(12,2),
`katharesPwlhseis` decimal(12,2),
`katharesPwlhseisXrisew` decimal(12,2),
`kostosPwlithentwn` decimal(12,2),
`ksenaKefalaia` decimal(12,2),
`kykloforounEnerghtiko` decimal(12,2),
`lhksiprothesmesYpoxrewseis` decimal(12,2),

```

```

`makroxroniesYpoxrewseis` decimal(12,2),
`mesoApothemaProiontwn` decimal(12,2),
`miktaKerdiEkmetallefsis` decimal(12,2),
`moEnerghtikou` decimal(12,2),
`moby` decimal(12,2),
`pagiaKefalaia` decimal(12,2),
`pagioEnerghtiko` decimal(12,2),
`provleponenesHmerisiesLeitourgikesDapanes` decimal(12,2),
`synolikaApasxoloumenaKefalaia` decimal(12,2),
`synolikaKefalaia` decimal(12,2),
`synolikoEnerghtiko` decimal(12,2),
`synoloIdiwnKefalaiwn` decimal(12,2),
`synoloKatavallomenwnMerismatwn` decimal(12,2),
`synoloKatharwnKerdwn` decimal(12,2),
`synoloMerismatwn` decimal(12,2),
`synoloTokwn` decimal(12,2),
`synoloYpoxrewsewn` decimal(12,2),
`timhMetoxhs` decimal(12,2),
`vraxyprothesmesYpoxrewseis` decimal(12,2),
`xrhmatisthriakhTimh` decimal(12,2),
`xrimatooikonomikaEksoda` decimal(12,2),
PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

/*Table structure for table `economics_economicuserinput` */

DROP TABLE IF EXISTS `economics_economicuserinput`;

CREATE TABLE `economics_economicuserinput` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `companyName` varchar(255) NOT NULL,
  `dateSaved` datetime NOT NULL,
  `miktaKerdiEkmetallefsis` decimal(12,2) DEFAULT NULL,
  `katharesPwlhseisXrisewn` decimal(12,2) DEFAULT NULL,
  `katharaKerdiEkmetallefsis` decimal(12,2) DEFAULT NULL,
  `xrimatooikonomikaEksoda` decimal(12,2) DEFAULT NULL,
  `synolikaApasxoloumenaKefalaia` decimal(12,2) DEFAULT NULL,
  `synolikoEnerghtiko` decimal(12,2) DEFAULT NULL,
  `synoloIdiwnKefalaiwn` decimal(12,2) DEFAULT NULL,
  `apodotikothtaIdiwnKefalaiwn` decimal(12,2) DEFAULT NULL,
  `apodotikothtaSynolouApasxoloumenwnKefalaiwn` decimal(12,2) DEFAULT NULL,
  `katharaKerdhProForwn` decimal(12,2) DEFAULT NULL,
  `moEnerghtikou` decimal(12,2) DEFAULT NULL,
  `idiaKefalaia` decimal(12,2) DEFAULT NULL,
  `synolikaKefalaia` decimal(12,2) DEFAULT NULL,
  `pagiaKefalaia` decimal(12,2) DEFAULT NULL,
  `makroxroniesYpoxrewseis` decimal(12,2) DEFAULT NULL,
  `ksenaKefalaia` decimal(12,2) DEFAULT NULL,
  `kykloforounEnerghtiko` decimal(12,2) DEFAULT NULL,
  `synoloYpoxrewsewn` decimal(12,2) DEFAULT NULL,
  `synoloTokwn` decimal(12,2) DEFAULT NULL,
  `katharesPwlhseis` decimal(12,2) DEFAULT NULL,
  `mesosOrosApaithsewn` decimal(12,2) DEFAULT NULL,
  `kostosPwlithentwn` decimal(12,2) DEFAULT NULL,
  `moby` decimal(12,2) DEFAULT NULL,
  `mesoApothemaProiontwn` decimal(12,2) DEFAULT NULL,
  `energhtikoArxhs` decimal(12,2) DEFAULT NULL,
  `energhtikoTelous` decimal(12,2) DEFAULT NULL,
  `pagioEnerghtiko` decimal(12,2) DEFAULT NULL,
  `synoloKatharwnKerdwn` decimal(12,2) DEFAULT NULL,
  `arithmosMetoxwn` decimal(12,2) DEFAULT NULL,

```

```

`synoloMerismatwn` decimal(12,2) DEFAULT NULL,
`arithmosMetoxwnSeKykloforia` decimal(12,2) DEFAULT NULL,
`merismaAnaMetoxh` decimal(12,2) DEFAULT NULL,
`timhMetoxhs` decimal(12,2) DEFAULT NULL,
`katharaKerdhXrhsewn` decimal(12,2) DEFAULT NULL,
`synoloKatavallomenwnMerismatwn` decimal(12,2) DEFAULT NULL,
`xrhmatisthriakhTimh` decimal(12,2) DEFAULT NULL,
`kerdhAnaMetoxh` decimal(12,2) DEFAULT NULL,
`diathesima` decimal(12,2) DEFAULT NULL,
`apaithseis` decimal(12,2) DEFAULT NULL,
`apothemata` decimal(12,2) DEFAULT NULL,
`vraxiprothesmesYpoxrewseis` decimal(12,2) DEFAULT NULL,
`diathesimoEnerghtiko` decimal(12,2) DEFAULT NULL,
`lhksiprothesmesYpoxrewseis` decimal(12,2) DEFAULT NULL,
`provlepomenesHmerisiesLeitourgikesDapanes` decimal(12,2) DEFAULT NULL,
`dateCreated` datetime NOT NULL,
PRIMARY KEY (`id`),
UNIQUE KEY `economics_economicsuserinput_companyName_2074aa11_uniq`
(`companyName`,`dateSaved`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

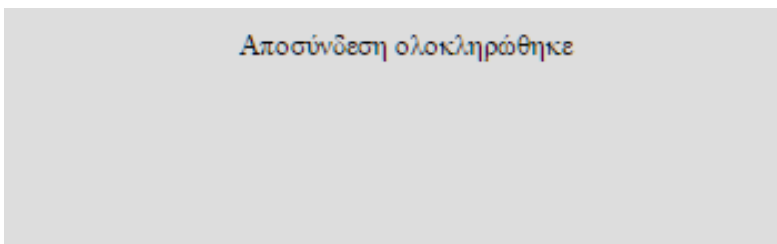
/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

```

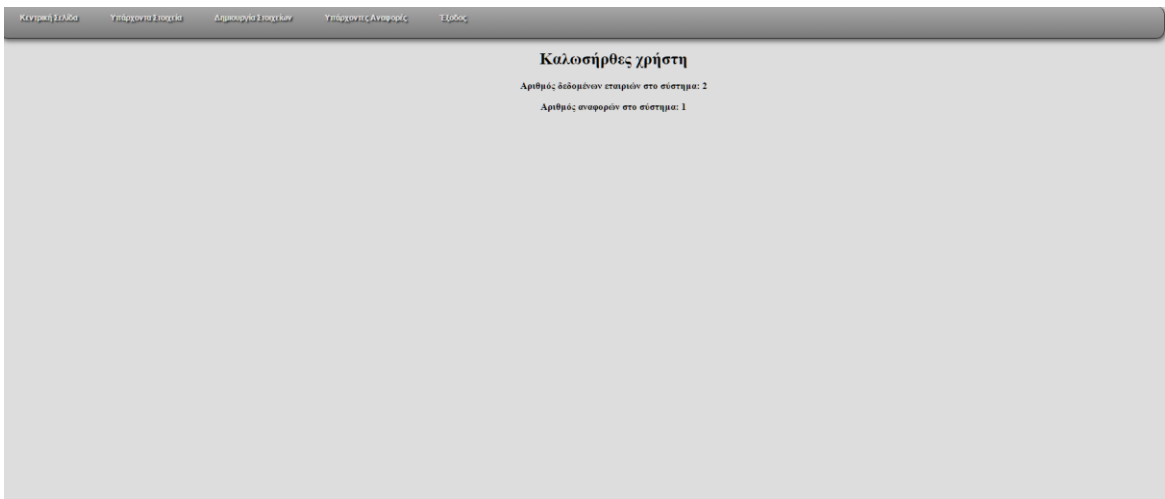
Παράρτημα Γ: Όψη οθονών εφαρμογής



Εικόνα 16 Είσοδος στο σύστημα



Εικόνα 17: Αποσύνδεση



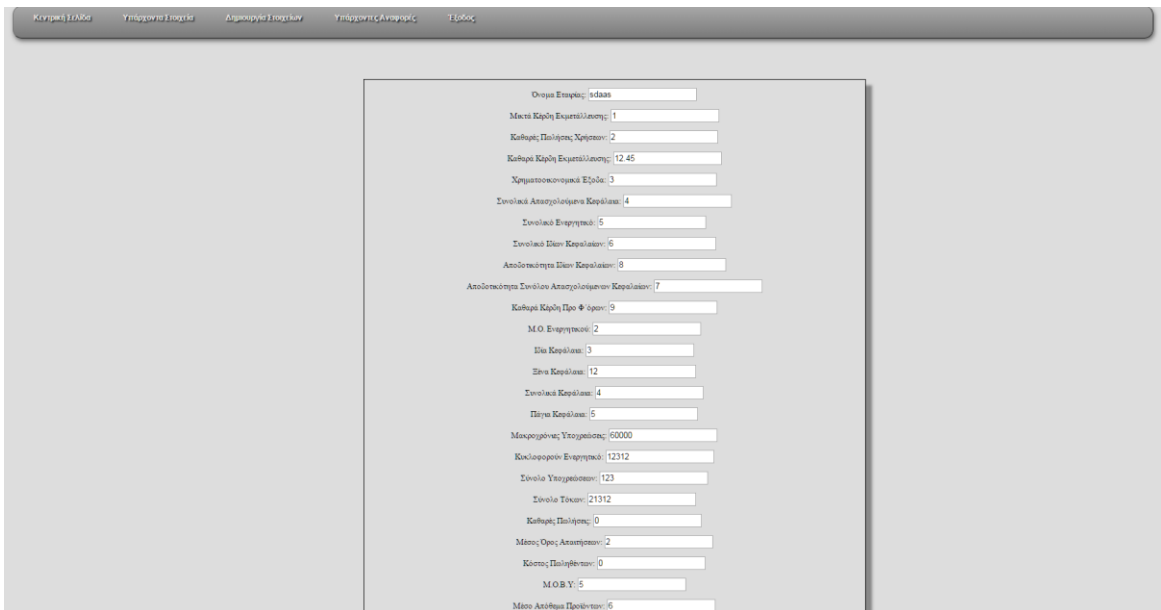
Εικόνα 18: Κεντρική Οθόνη Εφαρμογής

| Όνομα Εταιρίας @ Ημερομηνία Δημιουργίας | Πότε αποθηκεύτηκε | Στοιχεία Εισόδου | Διαγραφή Στοιχείων | Κλωνοποίηση | Δημιουργία Αναφοράς |
|---|--------------------------|-------------------|--------------------|-------------|---------------------|
| sdaas @ Aug. 12, 2015, 4:55 p.m. | Aug. 19, 2015, 5:39 p.m. | Προβολή Στοιχείων | Διαγραφή | Κλωνοποίηση | Αναφορά |
| sdaas @ Aug. 12, 2015, 5:17 p.m. | Aug. 12, 2015, 5:34 p.m. | Προβολή Στοιχείων | Διαγραφή | Κλωνοποίηση | Αναφορά |

Εικόνα 19: Υπάρχουσες αναφορές

| Υπάρχοντα Στοιχεία | Δημιουργία Στοιχείων | Υπάρχοντες Αναφορές | Έξοδος |
|---|----------------------|---------------------|---|
| Όνομα Εταιρίας: sdaas | | | Μικτά Κέρδη Εκμετάλλευσης: 1 |
| Καθαρές Πωλήσεις Χρήσεων: 2 | | | Καθαρά Κέρδη Εκμετάλλευσης: 12.45 |
| Χρηματοοικονομικά Έξοδα: 3 | | | Συνολικά Απασχολούμενα Κεφάλαια: 4 |
| Συνολικά Ενεργητικά: 5 | | | Συνολικά Ίδιων Κεφαλαίων: 6 |
| Αποδοτικότητα Ίδιων Κεφαλαίων: 8 | | | Αποδοτικότητα Συνόλου Απασχολούμενων Κεφαλαίων: 7 |
| Καθαρά Κέρδη Προ Φ' όρων: 9 | | | Μ.Ο. Ενεργητικού: 2 |
| Ίδια Κεφάλαια: 3 | | | Συνολικά Κεφάλαια: 4 |
| Πάγια Κεφάλαια: 5 | | | Μακροχρόνιες Υποχρεώσεις: 60000 |
| Ξένα Κεφάλαια: 12 | | | Κυκλοφορούν Ενεργητικό: 12312 |
| Σύνολο Υποχρεώσεων: 123 | | | Σύνολο Τάκων: 21312 |
| Καθαρές Πωλήσεις: 0 | | | Μέσος Όρος Απαιτήσεων: |
| Κόστος Πωληθέντων: 0 | | | Μ.Ο.Β.Υ: 5 |
| Μέσο Απόθεμα Προϊόντων: 6 | | | Ενεργητικό Αρχής: 7 |
| Ενεργητικό Τέλους: 9 | | | Πάγιο Ενεργητικό: 8 |
| Σύνολο Καθαρών Κερδών: 12 | | | Αριθμός Μετοχών: 321 |
| Σύνολο Μερισιμάτων: 312 | | | Αριθμός Μετοχών σε Κυκλοφορία: 312 |
| Μέρισμα Ανά Μετοχή: 321 | | | Τιμή Μετοχής: 43 |
| Καθαρά Κέρδη Χρήσεων: 235 | | | Σύνολο Καταβαλλόμενων Μερισιμάτων: 654 |
| Χρηματιστηριακή Τιμή: 756 | | | Κέρδη Ανά Μετοχή: 321 |
| Διαθέσιμα: 11 | | | Απαιτήσεις: 1 |
| Αποθέματα: 2 | | | Βραχυπρόθεσμες Υποχρεώσεις: 3 |
| Διαθέσιμο Ενεργητικό: 4 | | | Ληξιπρόθεσμες Υποχρεώσεις: 5 |
| Προβλεπόμενες Ημερήσιες Λειτουργικές Δαπάνες: 6 | | | |

Εικόνα 20 Προβολή στοιχείων εισόδου



Εικόνα 21 Επεξεργασία στοιχείων εισόδου (μέρος)

Κεντρική ΠΑΑΑ Υπάρχοντα Στοιχεία Δημιουργία Στοιχείων Υπάρχοντες Αναφορές Έξοδος

Όνομα Εταιρίας:

Μικτά Κέρδη Εκμετάλλευσης:

Καθαρά Πωλήσεις Χρήσεων:

Καθαρά Κέρδη Εκμετάλλευσης:

Χρηματοοικονομικά Έσοδα:

Συνολικά Απασχολούμενα Κεφάλαια:

Συνολικό Ενεργητικό:

Συνολικό Ίδιον Κεφάλαιον:

Αποδοτικότητα Ίδιον Κεφαλαίων:

Αποδοτικότητα Συνόλου Απασχολούμενων Κεφαλαίων:

Καθαρά Κέρδη Προ Φόρων:

Μ.Ο. Ενεργητικού:

Ίδια Κεφάλαια:

Ξένα Κεφάλαια:

Συνολικά Κεφάλαια:

Πάγια Κεφάλαια:

Μακροχρόνιες Υποχρεώσεις:

Κυκλοφορούν Ενεργητικό:

Σύνολο Υποχρεώσεων:

Σύνολο Τόκων:

Καθαρά Πωλήσεις:

Μέσος Όρος Απαιτήσεων:

Κόστος Πωληθέντων:

Μ.Ο.Β.Υ:

Μέσο Απόθεμα Προϊόντων:

Εικόνα 22 Εισαγωγή Στοιχείων (μέρος)

| Όνομα Εταιρίας / Ημερομηνία Δημιουργίας | Πότε δημιουργήθηκε | Στοιχεία Εισόδου | Προβολή Αναφοράς | Διαγραφή Αναφοράς |
|---|--------------------------|-------------------|------------------|-------------------|
| sdaas @ Aug. 12, 2015, 4:55 p.m. | Aug. 19, 2015, 5:44 p.m. | Προβολή Στοιχείων | Προβολή | Διαγραφή |

Εικόνα 23 Υπάρχουσες αναφορές

Υπάρχοντα Στοιχεία Δημιουργία Στοιχείων Υπάρχοντες Αναφορές Έξοδος

| | |
|---|---|
| Όνομα Εταιρίας: sdaas | Μικτά Κέρδη Εκμετάλλευσης: 1 |
| Καθαρές Πωλήσεις Χρήσεων: 2 | Καθαρά Κέρδη Εκμετάλλευσης: 12.45 |
| Χρηματοοικονομικά Έσοδα: 3 | Συνολικά Απασχολούμενα Κεφάλαια: 4 |
| Συνολικό Ενεργητικό: 5 | Συνολικό Ίδιον Κεφάλαιον: 6 |
| Αποδοτικότητα Ίδιον Κεφαλαίων: 8 | Αποδοτικότητα Συνόλου Απασχολούμενων Κεφαλαίων: 7 |
| Καθαρά Κέρδη Προ Φόρων: 9 | Μ.Ο. Ενεργητικού: 2 |
| Ίδια Κεφάλαια: 3 | Συνολικά Κεφάλαια: 4 |
| Πάγια Κεφάλαια: 5 | Μακροχρόνιες Υποχρεώσεις: 60000 |
| Ξένα Κεφάλαια: 12 | Κυκλοφορούν Ενεργητικό: 12312 |
| Σύνολο Υποχρεώσεων: 123 | Σύνολο Τόκων: 21312 |
| Καθαρά Πωλήσεις: 0 | Μέσος Όρος Απαιτήσεων: |
| Κόστος Πωληθέντων: 0 | Μ.Ο.Β.Υ: 5 |
| Μέσο Απόθεμα Προϊόντων: 6 | Ενεργητικό Αρχής: 7 |
| Ενεργητικό Τέλους: 9 | Πάγια Ενεργητικό: 8 |
| Σύνολο Καθαρών Κερδών: 12 | Αριθμός Μετοχών: 321 |
| Σύνολο Μερισμάτων: 312 | Αριθμός Μετοχών σε Κυκλοφορία: 312 |
| Μέρισμα Ανά Μετοχή: 1 | Τιμή Μετοχής: 43 |
| Καθαρά κέρδη Χρήσεων: 235 | Σύνολο Καταβαλλόμενων Μερισμάτων: 654 |
| Χρηματιστηριακή Τιμή: 756 | Κέρδη Ανά Μετοχή: 0.04 |
| Διαθέσιμα: 11 | Απαιτήσεις: 1 |
| Αποθέματα: 2 | Βραχυπρόθεσμες Υποχρεώσεις: 3 |
| Διαθέσιμο Ενεργητικό: 4 | Ληξιπρόθεσμες Υποχρεώσεις: 5 |
| Προβλεπόμενες Ημερήσιες Λειτουργικές Δαπάνες: 6 | |

Εικόνα 24 Προβολή στοιχείων αναφοράς

| Εταιρία: sdaas | | | |
|---|----------|---|--------|
| Λίστες Αποδοτήτων | | | |
| Αριθμοδείκτη Μεταύ Περιβαλίου | 50 | Αποδοτήριο Αποσπορευμένων Κεφαλαίων | 388,25 |
| P.O.A. | 3,09 | P.O.E. | 2,08 |
| Αριθμοδείκτη Οικονομικής Μέλισης | 1,14 | | |
| Λίστρες Αξίετες | | | |
| P.O.A. (B τρέτες) | 4,5 | P.O.L. | 0,4 |
| Λίστες Αιρέθρατες Και Βιοαυτέτες | | | |
| Αριθμοδείκτη Ιδίων Προς Σικολιά Κεφάλαια | 75 | Αριθμοδείκτη Ιδίων Προς Είκο Κεφάλαια | 0,25 |
| Αριθμοδείκτη Ιδίων Προς Πάγο Κεφάλαια | 0,0 | Αριθμοδείκτη Παγών Κεφαλαίων Προς Μορφωμένες Υπαρξίετες | 0 |
| Αριθμοδείκτη Κυκλοφορούντες Ενάγνητοί Προς Σικολιά Υπαρξίετες | 10009,70 | Αριθμοδείκτη Κάλυψης Τόκων | 0 |
| Λίστες Αρρατεμείετες | | | |
| Αριθμοδείκτη Ταχίετες Επισποής Απαίεσεων | 0 | Αριθμοδείκτη Ταχίετες Βραχυπρόθεσμων Υπαρξίετων | 0 |
| Αριθμοδείκτη Ταχίετες Κυκλοφορίας Αποθεμείων | 0 | Μ.Ο. Εκερητηνού | 11,5 |
| Αριθμοδείκτη Ταχίετες Κυκλοφορίας Καθαρού Κεφαλαίου | 0 | Αριθμοδείκτη Ταχίετες Κυκλοφορίας Ιδίων Κεφαλαίων | 0 |
| Αριθμοδείκτη Ταχίετες Κυκλοφορίας Παγών Κεφαλαίων | 0 | | |
| Επενδυτικοί Λίστες | | | |
| Κέρη ανά Μεταχί | 0,04 | Μέρομα ανά Μεταχί | 1 |
| Μορματητή Απέδοση | 2,33 | Αριθμοδείκτη Κάλυψης Καταβαλλόμενων Μορματίων | 0,30 |
| Ευκερητή Αξία Μεταχί | 0,02 | Π προς Ε.Π.Ε. | 20223 |
| Λίστες Ρευστόετες | | | |
| Αριθμοδείκτη Γενίης Ρευστόετες | 4,07 | Αριθμοδείκτη Είδικής Ρευστόετες | 4 |
| Αριθμοδείκτη Ταμειακής Ρευστόετες | 0,8 | Αριθμοδείκτη Αμυντικοί Χρονηοί Διαστήετες | 730 |

Εικόνα 25 Προβολή Αναφοράς

Βιβλιογραφία

Νικήτα Α. Νιάρχου Χρηματοοικονομική ανάλυση λογιστικών καταστάσεων Αθήνα 2004,
Εκδόσεις Αθ. Σταμούλης

Ιστοσελίδες(websites)

JAVA:

[https://en.wikipedia.org/wiki/Java_\(programming_language\)](https://en.wikipedia.org/wiki/Java_(programming_language))

<https://www.oracle.com/java/index.html>

https://java.com/en/download/faq/whatis_java.xml

PYTHON:

<https://www.python.org/>

[https://en.wikipedia.org/wiki/Python_\(programming_language\)](https://en.wikipedia.org/wiki/Python_(programming_language))

<http://blog.teamtreehouse.com/what-is-python>

PYTHON DJANGO

<https://www.djangoproject.com/>

<https://docs.djangoproject.com/en/1.8/intro/tutorial01/>

[https://en.wikipedia.org/wiki/Django_\(web_framework\)](https://en.wikipedia.org/wiki/Django_(web_framework))

<https://www.airpair.com/python/posts/django-flask-pyramid>

PYTHON VERSUS JAVA

<https://pythonconquerstheuniverse.wordpress.com/2009/10/03/python-java-a-side-by-side-comparison/>

<https://blog.udemy.com/python-vs-java/>

MySQL

<https://www.mysql.com/>

<https://en.wikipedia.org/wiki/MySQL>

<http://www.cio.com/article/2438900/data-management/five-compelling-reasons-to-use-mysql.html>

PostgreSQL

<http://www.postgresql.org/>

<https://en.wikipedia.org/wiki/PostgreSQL>

<http://2ndquadrant.com/en/postgresql/who-uses-postgresql/>

MySQL versus PostgreSQL

https://www.wikivs.com/wiki/MySQL_vs_PostgreSQL

<https://www.digitalocean.com/community/tutorials/sqlite-vs-mysql-vs-postgresql-a-comparison-of-relational-database-management-systems>

<http://www.enterisedb.com/solutions/mysql-vs-postgresql>

<https://www.quora.com/What-are-pros-and-cons-of-PostgreSQL-and-MySQL>