



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ Μ/Υ  
ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ  
ΣΧΟΛΗ ΝΑΥΤΙΛΙΑΣ ΚΑΙ ΒΙΟΜΗΧΑΝΙΑΣ  
ΤΜΗΜΑΤΟΣ ΒΙΟΜΗΧΑΝΙΚΗΣ ΔΙΟΙΚΗΣΗΣ & ΤΕΧΝΟΛΟΓΙΑΣ  
ΔΙΑΠΑΝΕΠΙΣΤΗΜΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ  
«ΤΕΧΝΟ-ΟΙΚΟΝΟΜΙΚΑ ΣΥΣΤΗΜΑΤΑ»



## **Σχεδιασμός και Ανάπτυξη Πλατφόρμας για την Καταχώρηση Δράσεων Εξοικονόμησης Ενέργειας Επιχειρήσεων**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

**Νικόλαου Ι. Σακελλαρίου**

Επιβλέπων: Χρυσόστομος Δούκας, Αναπληρωτής Καθηγητής ΕΜΠ

Αθήνα, Φεβρουάριος 2021



Copyright © Νικόλαος Ι. Σακελλαρίου, 2021

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν το συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

## ***Ευχαριστίες***

*Πρώτα απ' όλα, θέλω να ευχαριστήσω τον επιβλέποντα της διπλωματικής μου εργασίας, Αναπληρωτή Καθηγητή κο Χρυσόστομο Δούκα, για την πολύτιμη υποστήριξη, βοήθεια και καθοδήγησή του κατά τη διάρκεια της δουλειάς μου.*

*Πάνω απ' όλα όμως, θέλω να εκφράσω την ευγνωμοσύνη μου στην οικογένειά μου και στα αγαπημένα μου πρόσωπα για όλη τη βοήθεια, υποστήριξη, εμπιστοσύνη και αντοχή που επέδειξαν κατά τη διάρκεια των σπουδών μου.*



## Πίνακας Περιεχομένων

Ευρετήριο Γραφημάτων .....	iii
Ευρετήριο Εικόνων.....	iv
Ευρετήριο Scripts .....	vi
Ευρετήριο Ακρωνυμίων.....	vii
Περίληψη .....	ix
Abstract.....	x
<b>1. ΕΙΣΑΓΩΓΗ .....</b>	<b>1</b>
1.1. Στόχος της Εργασίας .....	2
1.2. Δομή του Εγγράφου .....	3
<b>2. ΤΟ ΤΕΧΝΟΛΟΓΙΚΟ ΥΠΟΒΑΘΡΟ ΑΝΑΠΤΥΞΗΣ ΤΗΣ ΠΛΑΤΦΟΡΜΑΣ.....</b>	<b>4</b>
2.1. Microsoft Visual Studio .....	4
2.2. Η πλατφόρμα .NET .....	5
2.2.1. Η αρχιτεκτονική του .NET Framework.....	5
2.2.1.1. LINQ .....	8
2.2.2. ASP.NET .....	9
2.2.2.1. ASP.NET Core .....	9
2.2.2.2. ASP.NET Core MVC .....	10
2.2.2.3. ASP.NET Core Identity.....	11
2.3. Η γλώσσα προγραμματισμού C# .....	12
2.4. HTML.....	14
2.5. CSS.....	14
2.6. JavaScript .....	15
2.7. jQuery.....	16
2.8. Ajax .....	17
2.9. MySQL.....	18
2.10. Entity Framework.....	19
<b>3. ΥΛΟΠΟΙΗΣΗ ΤΗΣ ΠΛΑΤΦΟΡΜΑΣ.....</b>	<b>21</b>
3.1. Υλοποίηση της βάσης δεδομένων .....	21

3.1.1. Δημιουργία των πινάκων .....	21
3.1.2. Δημιουργία του View ηρν .....	26
3.1.3. Υλοποίηση αυτόματης εισαγωγής δεδομένων στον πίνακα payments .....	27
3.1.4. Υλοποίηση αυτόματης πρόκρισης της κατάστασης των δράσεων.....	29
3.1.5. Διάγραμμα ER της βάσης δεδομένων .....	31
3.2. Δόμηση της εφαρμογής .....	32
3.2.1. Εγγραφή και είσοδος χρηστών .....	34
3.2.2. Υλοποίηση του προτύπου MVC.....	35
3.2.2.1. Routing.....	36
3.2.2.2. Controllers.....	36
3.2.2.3. Models.....	39
3.2.2.4. Views .....	42
<b>4. ΠΑΡΟΥΣΙΑΣΗ ΤΟΥ UI ΚΑΙ ΤΗΣ ΛΕΙΤΟΥΡΓΙΑΣ ΤΗΣ ΠΛΑΤΦΟΡΜΑΣ .....</b>	<b>44</b>
4.1. Αρχική σελίδα .....	44
4.2. Σύνδεση Χρηστών .....	46
4.3. Εγγραφή Χρηστών .....	47
4.3.1. Καταχώρηση Επιχείρησης.....	47
4.3.2. Καταχώρηση Επενδυτή .....	52
4.4. Σελίδα καταχωρημένων δράσεων .....	57
4.5. Σελίδα καταχώρησης νέας δράσης.....	58
4.6. Σελίδα ανάλυσης δράσης .....	62
4.7. Προφίλ Χρηστών.....	66
4.7.1. Προφίλ Επιχείρησης.....	66
4.7.2. Προφίλ Επενδυτή .....	70
<b>5. ΣΥΜΠΕΡΑΣΜΑΤΑ .....</b>	<b>74</b>
<b>6. ΒΙΒΛΙΟΓΡΑΦΙΚΕΣ ΑΝΑΦΟΡΕΣ .....</b>	<b>76</b>

## Ευρετήριο Γραφημάτων

Διάγραμμα 1: Η Αρχιτεκτονική του . NET Framework .....	7
Διάγραμμα 2: Οι τεχνολογίες του LINQ .....	9
Διάγραμμα 3: Απόδοση του ASP.NET Core .....	10
Διάγραμμα 4: Η αρχιτεκτονική του ASP.NET Core Identity .....	12
Διάγραμμα 5: Η λειτουργία του Ajax .....	18
Διάγραμμα 6: Η αρχιτεκτονική του Entity Framework .....	20
Διάγραμμα 7: ER Diagram .....	32



## Ευρετήριο Εικόνων

Εικόνα 1: Πίνακας user.....	23
Εικόνα 2: Πίνακας business.....	23
Εικόνα 3: Πίνακας investor.....	24
Εικόνα 4: Πίνακας action.....	25
Εικόνα 5: Πίνακας actionsegment .....	25
Εικόνα 6: Πίνακας payments .....	26
Εικόνα 7: Δημιουργία Μοντέλων μέσω του EF .....	33
Εικόνα 8: Αρχική σελίδα .....	45
Εικόνα 9: Σελίδα σύνδεσης.....	46
Εικόνα 10: Βοηθητική σελίδα εγγραφής χρηστών .....	47
Εικόνα 11: Εγγραφή Επιχείρησης - Γενικά Στοιχεία.....	48
Εικόνα 12: Εγγραφή Επιχείρησης - Email.....	49
Εικόνα 13: Εγγραφή Επιχείρησης - Διαπιστευτήρια .....	49
Εικόνα 14: Εγγραφή Επιχείρησης - Διεύθυνση.....	50
Εικόνα 15: Εγγραφή Επιχείρησης - Α.Φ.Μ.....	51
Εικόνα 16: Εγγραφή Επιχείρησης - IBAN .....	51
Εικόνα 17: Εγγραφή Επιχείρησης - Νόμιμος Εκπρόσωπος .....	52
Εικόνα 18: Εγγραφή Επενδυτή - Ονοματεπώνυμο.....	53
Εικόνα 19: Εγγραφή Επενδυτή - Email .....	54
Εικόνα 20: Εγγραφή Επενδυτή - Διαπιστευτήρια.....	54
Εικόνα 21: Εγγραφή Επενδυτή – Διεύθυνση.....	55
Εικόνα 22: Εγγραφή Επενδυτή – Υπηκοότητα.....	56
Εικόνα 23: Εγγραφή Επιχείρησης - Α.Φ.Μ.....	56
Εικόνα 24: Σελίδα καταχωρημένων δράσεων .....	58

Εικόνα 25: Καταχώρηση νέας δράσης - Τίτλος.....	59
Εικόνα 26: Καταχώρηση νέας δράσης - Κατηγορία.....	60
Εικόνα 27: Καταχώρηση νέας δράσης - Περιγραφή .....	61
Εικόνα 28: Καταχώρηση νέας δράσης - Πληροφορίες.....	62
Εικόνα 29: Σελίδα δράσης – Γενική Απεικόνιση .....	63
Εικόνα 30: Σελίδα δράσης – Απεικόνιση Επενδυτή.....	64
Εικόνα 31: Προφίλ Επιχείρησης – Στοιχεία Λογαριασμού.....	67
Εικόνα 32: Προφίλ Επιχείρησης - Καταχωρημένες Δράσεις .....	68
Εικόνα 33: Προφίλ Επιχείρησης – Οικονομική Επισκόπηση.....	69
Εικόνα 34: Προφίλ Επενδυτή - Στοιχεία Λογαριασμού .....	71
Εικόνα 35: Προφίλ Επενδυτή – Συμμετοχές σε Δράσεις .....	72
Εικόνα 36: Προφίλ Επενδυτή – Οικονομική Επισκόπηση .....	73

## Ευρετήριο Scripts

Script 1: Δημιουργία ηρν View .....	27
Script 2: Δημιουργία Trigger actionsegment_AFTER_INSERT .....	28
Script 3: Δημιουργία Procedure create_payment_periods.....	28
Script 4: Δημιουργία events στον scheduler .....	30
Script 5: Δημιουργία Procedure completed_actions.....	30
Script 6: Δημιουργία Procedure canceled_actions.....	31
Script 7: Χρήση του Entity Framework.....	33
Script 8: Διαμόρφωση της κλάσης Startup .....	35
Script 9: Model ActionRegister .....	39
Script 10: Model ActionSearch.....	39
Script 11: Model ActionView .....	40
Script 12: Model BusinessProfile .....	40
Script 13: Model BusinessUserRegister .....	40
Script 14: Model InvestorProfile .....	41
Script 15: Model InvestorUserRegister .....	41
Script 16: Model CountriesSectorsList.....	42
Script 17: Login Profile .....	42

## Ευρετήριο Ακρωνυμίων

<b>ΑΠΕ</b>	Ανανεώσιμες Πηγές Ενέργειας
<b>ΕΕ</b>	Ευρωπαϊκή Ένωση
<b>ΕΕ-27</b>	Ευρωπαϊκή Ένωση των 27 Κρατών Μελών
<b>ADO</b>	ActiveX Data Object
<b>AJAX</b>	Asynchronous JavaScript and XML
<b>API</b>	Application Programming Interface
<b>ASP</b>	Active Server Pages
<b>CLI</b>	Common Language Infrastructure
<b>CLR</b>	Common Language Runtime
<b>COM</b>	Component Object Model
<b>CSS</b>	Cascading Style Sheets
<b>DB</b>	DataBase
<b>DOM</b>	Document Object Model
<b>EF</b>	Entity Framework
<b>FK</b>	Foreign Key
<b>GJ</b>	Giga Joule
<b>HTML</b>	HyperText Markup Language
<b>IBAN</b>	International Bank Account Number
<b>IDE</b>	Integrated Development Environment
<b>IoT</b>	Internet of Things
<b>JIT</b>	Just In Time
<b>JSON</b>	JavaScript Object Notation
<b>LINQ</b>	Language Integrated Query
<b>MVC</b>	Model View Controller
<b>NPV</b>	Net Present Value
<b>ORM</b>	Object–Relational Mapping
<b>PK</b>	Primary Key
<b>RDBMS</b>	Relational Database Management System
<b>SQL</b>	Structured Query Language
<b>tCO<sub>2</sub>e</b>	Tons of Carbon Dioxide Equivalent

<b>UI</b>	User Interface
<b>URL</b>	Uniform Resource Locator
<b>VS</b>	Visual Studio
<b>WCF</b>	Windows Communication Foundation
<b>WPF</b>	Windows Presentation Foundation
<b>XHR</b>	XML Http Request
<b>XML</b>	eXtensible Markup Language

## Περίληψη

Η χρήση των υδρογονανθράκων εντοπίζεται ήδη από το 19<sup>ο</sup> αιώνα, αρχικά για φωτισμό και θέρμανση και μετέπειτα στις μηχανές εσωτερικής καύσης. Η βιομηχανική επανάσταση οδήγησε στην κατακόρυφη αύξηση της χρήσης τους καθιστώντας τη σύγχρονη οικονομία, εξαρτημένη από την ανάλωση ορυκτών καυσίμων. Ωστόσο, πρόκειται για μη ανανεώσιμες πηγές ενέργειας που απαιτούν εκατομμύρια χρόνια για να σχηματιστούν, ενώ παράλληλα η καύση τους συνεπάγεται την εκπομπή αέριων ρύπων που εντείνουν το φαινόμενο του θερμοκηπίου και ελλοχεύουν κινδύνους για το περιβάλλον. Καθίσταται λοιπόν σαφές, η ανάγκη για περιορισμό της σπατάλης των ενεργειακών αποθεμάτων και κατ' επέκταση για ανεξάρτηση από τη χρήση μη ανανεώσιμων ενεργειακών πόρων.

Στα πλαίσια της προσέγγισης αυτής, η παρούσα μελέτη παρουσιάζει μία πλατφόρμα για τις επιχειρήσεις που επιθυμούν τη χρηματοδότηση δράσεων εξοικονόμησης ενέργειας. Ειδικότερα, περιλαμβάνει την καταχώρηση τέτοιων δράσεων με στόχο την προσέλκυση ιδιωτικών επενδύσεων. Στόχος της είναι η προώθηση δράσεων που θα οδηγήσουν σε βελτίωση της ενεργειακής απόδοσης κτιρίων, εγκαταστάσεων, μηχανημάτων και επιχειρηματικών διαδικασιών, καθώς και σε αύξηση της παραγωγής και ανάλωσης ενέργειας από ανανεώσιμες πηγές στο βιομηχανικό τομέα. Με τον τρόπο αυτό επιτυγχάνεται η περιστολή της σπατάλης των περιορισμένων ενεργειακών αποθεμάτων του πλανήτη υποστηρίζοντας παράλληλα τη βιωσιμότητα και την ανάπτυξη της οικονομίας.

Πρόκειται για μια διαδικτυακή εφαρμογή, η κατασκευή της οποίας έχει υλοποιηθεί με την σύμμειξη μιας πληθώρας σύγχρονων τεχνολογιών. Η συγκεκριμένη πλατφόρμα έχει βελτιστοποιηθεί ως προς την ταχύτητα, την ευκολία χρήσης αλλά και την διεπαφή που προσφέρει στους επισκέπτες της.

## Λέξεις Κλειδιά

Πλατφόρμα, Εξοικονόμηση Ενέργειας, Ενεργειακή Απόδοση, Ανανεώσιμες Πηγές Ενέργειας, Χρηματοδότηση Ενέργειας

## Abstract

The use of hydrocarbons dates back to the 19th century, first for lighting and heating and later in internal combustion engines. The industrial revolution led to a sharp increase in their use, making the modern economy dependent on fossil fuel consumption. However, these are non-renewable energy sources that take millions of years to form, while their combustion involves the emission of gaseous pollutants that intensify the greenhouse effect and pose a risk to the environment. Thus, the need to reduce the waste of energy reserves and consequently to become independent of the use of non-renewable energy resources becomes clear.

As part of this approach, this study presents a platform for companies seeking to fund energy saving actions. In particular, it includes the registration of such actions in order to attract private sector investments. Its aim is to promote actions that will lead to improved energy efficiency of buildings, facilities, machinery and business processes, as well as to increased production and consumption of energy from renewable sources in the industrial sector. In this way, achieves the reduction of waste of the planet's limited energy reserves, while supporting sustainability and economic growth.

It is a web-based application, the development of which has been implemented by mixing a variety of modern technologies. This platform has been optimized in terms of speed, ease of use and user interface (UI).

## Keywords

Platform, Energy Saving, Energy Efficiency, Renewable Energy Sources, Energy Finance

## 1. ΕΙΣΑΓΩΓΗ

Η χρήση των ορυκτών καυσίμων για την παραγωγή ενέργειας ήταν αυτή που οδήγησε στη βιομηχανική επανάσταση και αναφέρεται μόλις στην πρώιμη ιστορία της ανθρωπότητας. Ωστόσο, η ανεξέλεγκτη χρήση τους μέχρι και σήμερα σε συνδυασμό με άλλους παράγοντες έχει οδηγήσει στο αποκαλούμενο «ενεργειακό ζήτημα» συνθέτοντας ένα μάλλον δυσοίωνο μέλλον για την ανθρωπότητα αλλά και τον ίδιο τον πλανήτη μας.

Η ραγδαία τεχνολογική ανάπτυξη ως απόρροια της βιομηχανικής επανάστασης σε συνδυασμό με τον ολοένα αυξανόμενο πληθυσμό της Γης έχει οδηγήσει σε αυξημένες απαιτήσεις σε ενεργειακούς πόρους, το μεγαλύτερο μέρος εκ των οποίων καλύπτεται από τον περιορισμένο ορυκτό πλούτο του πλανήτη μας. Η χρήση όμως των ορυκτών καυσίμων είναι η αιτία της έντασης μιας πληθώρας περιβαλλοντικών ζητημάτων, που ελλοχεύουν κινδύνους τόσο για τον ίδιο τον άνθρωπο όσο και για το μέλλον αυτού του πλανήτη. Για την αντιμετώπιση, λοιπόν, του ενεργειακού προβλήματος απαιτείται τόσο ο περιορισμός της χρήσης ενεργειακών πόρων μέσω μεθόδων αποδοτικότερης αξιοποίησης και περιορισμού της κατασπατάλησης αυτών όσο και η ανάπτυξη νέων ήπιων πηγών ενέργειας, όπως η υδροηλεκτρική, αιολική, ηλιακή, βιομάζα κ.ά. Αυτός είναι και ο ορισμός της εξοικονόμησης ενέργειας και γενικότερα οποιαδήποτε προσπάθεια με την οποία μπορεί να επιτευχθεί ο περιορισμός της σπατάλης των ενεργειακών αποθεμάτων.

Με έμφαση στην κλιματική αλλαγή και την υποβάθμιση του περιβάλλοντος η ΕΕ έχει αναπτύξει μια στρατηγική για την αντιμετώπιση αυτών των προκλήσεων, μέσω της προώσπισης της χρήσης με σύγχρονο και αποδοτικό τρόπο των πόρων της. Κύριος στόχος της είναι η εκμηδένιση των καθαρών εκπομπών αερίων του θερμοκηπίου μέχρι το 2050 και για το λόγο αυτό έχει προχωρήσει στην υιοθέτηση της Ευρωπαϊκής Πράσινη Συμφωνίας, η οποία παρέχει ένα σχέδιο δράσης για την ενίσχυση της αποδοτικής χρήσης των πόρων με τη μετάβαση σε μια καθαρή, κυκλική οικονομία και για την αποκατάσταση της βιοποικιλότητας και τη μείωση της ρύπανσης. [1]

Στον δρόμο αυτό, με οδηγίες και χάρτες πορείας η ΕΕ έχει θέσει ορόσημα για τα έτη 2030, 2040 και 2050 θέτοντας μέσα από ενεργειακές στρατηγικές, στόχους για την αύξηση τόσο της ενεργειακής απόδοσης όσο και του μεριδίου των ΑΠΕ στο ενεργειακό της μείγμα.



Ειδικότερα, μέχρι το 2030 η ΕΕ στοχεύει σε ενεργειακή απόδοση τουλάχιστον 32,5% ενώ το μερίδιο ΑΠΕ θα πρέπει να ανέρχεται σε ποσοστό 32% κατ' ελάχιστο. [2]

### 1.1. Στόχος της Εργασίας

Αντικείμενο της παρούσας μελέτης αποτελεί η ανάπτυξη μιας πλατφόρμας που μπορεί να χρησιμεύσει όχι μόνο ως αποθετήριο δράσεων εξοικονόμησης ενέργειας αλλά και ως ένα εργαλείο στα χέρια επιχειρήσεων και επενδυτών. Στόχος της πλατφόρμας είναι η ενίσχυση της ιδιωτικής πρωτοβουλίας για την ανάληψη έργων που θα οδηγήσουν σε μείωση της κατανάλωσης ενέργειας στη βιομηχανία.

Συγκεκριμένα, μέσω της βελτιστοποίησης της αποδοτικής χρήσης ενεργειακών πόρων μειώνεται το ενεργειακό κόστος των επιχειρήσεων και κατ' επέκταση το κόστος παραγωγής εντείνοντας την ανταγωνιστικότητα μεταξύ τους. Η εξοικονόμηση αυτή ενέργειας, που μπορεί να προκύψει για παράδειγμα από αντικατάσταση καυστήρων, θερμομόνωση αγωγών και κτιρίων ή ακόμη και εγκατάσταση εναλλακτικών συστημάτων ΑΠΕ, οδηγεί σε μείωση των εκπομπών των αέριων ρύπων, μείωση του ενεργειακού ισοζυγίου της χώρας και κατ' επέκταση της ΕΕ, αλλά και σταδιακή απεξάρτηση από την εισαγωγή υδρογονανθράκων.

Η πλατφόρμα προσφέρει ένα απλό και εύχρηστο τρόπο για τις επιχειρήσεις να αναζητήσουν ιδιωτικά κεφάλαια για την προώθηση δράσεων εξοικονόμησης ενέργειας. Με τον τρόπο αυτό φέρνει σε επαφή επενδυτές και επιχειρήσεις μέσω της χρηματοδότησης αυτών των δράσεων, οι οποίες δημιουργούν ευκαιρίες και οφέλη και στα δύο μέρη. Πρόκειται για έναν εναλλακτικό τρόπο χρηματοδότησης τέτοιων δράσεων πέραν των χρηματοπιστωτικών ιδρυμάτων, που δύναται να χρησιμεύσει ακόμη και ως εργαλείο προώθησης των προαναφερθέντων ενεργειακών στόχων της ΕΕ.

Σκοπός της παρούσας εργασίας είναι η ανάδειξη ενός εργαλείου για την προώθηση των δράσεων εξοικονόμησης ενέργειας στον ιδιωτικό τομέα. Αν και οι δυνατότητες της πλατφόρμας είναι περιορισμένες στις σχεδόν απαραίτητες στα πλαίσια των αναγκών της παρούσας μελέτης, ωστόσο είναι επεκτάσιμες και δύναται να ικανοποιήσουν τις ανάγκες μιας πλήρους λειτουργικής εφαρμογής. Με τη χρήση μιας τέτοιας εφαρμογής υπάρχει η δυνατότητα ώθησης προς ανάληψη νέων τέτοιων δράσεων, που θα τυγχάνουν ιδιωτικής χρηματοδότησης προσφέροντας πλεονεκτήματα τόσο στις ίδιες τις επιχειρήσεις όσο και στους επενδυτές.

## 1.2. Δομή του Εγγράφου

Το δεύτερο κεφάλαιο πραγματεύεται τις τεχνολογίες που αξιοποιήθηκαν στα πλαίσια αυτής της εργασίας με στόχο την ανάπτυξη της πλατφόρμας. Ειδικότερα, γίνεται ανάλυση όλων των τεχνολογιών που χρησιμοποιήθηκαν τόσο σε επίπεδο διεπαφής όσο και σε επίπεδο λογικής της εφαρμογής και ενσωμάτωσης της διεπαφής σε αυτή.

Στο τρίτο κεφάλαιο γίνεται αναφορά στην υλοποίηση της πλατφόρμας, δηλαδή στον τρόπο δόμησης της εφαρμογής και της βάσης δεδομένων. Συγκεκριμένα, χρησιμοποιήθηκε το Visual Studio 2019 της Microsoft για τη συγγραφή κώδικα σε γλώσσα C# με την υιοθέτηση του προτύπου ASP.NET Core MVC. Για την κατασκευή του UI της πλατφόρμας έγινε χρήση HTML, CSS, JavaScript, jQuery και Ajax, ενώ η βάση δεδομένων της δημιουργήθηκε σε MySQL με τη βοήθεια του MySQL Workbench. Η δημιουργία των μοντέλων της βάσης δεδομένων αλλά και η σύνδεση της εφαρμογής με την τελευταία στα πλαίσια του προτύπου MVC πραγματοποιήθηκε μέσω του EF. Τέλος, η πρόσβαση στην εφαρμογή, η δημιουργία και η διαχείριση χρηστών υλοποιήθηκε με τη βοήθεια του ASP.NET Core Identity.

Στο τέταρτο κεφάλαιο γίνεται μια εκτενής αναφορά της διεπαφής και του τρόπου λειτουργίας της πλατφόρμας καθώς και της πλοήγησης στις ιστοσελίδες της. Πραγματοποιείται μια παρουσίαση των βασικών ιστοσελίδων και του μενού της εφαρμογής, αλλά και ο τρόπος δημιουργίας λογαριασμών και της σύνδεσης των χρηστών σε αυτή. Ακόμη, αναφέρεται η μέθοδος καταχώρησης νέων δράσεων και γίνεται ανάλυση της σελίδας του προφίλ για κάθε είδος χρήστη.

## 2. ΤΟ ΤΕΧΝΟΛΟΓΙΚΟ ΥΠΟΒΑΘΡΟ ΑΝΑΠΤΥΞΗΣ ΤΗΣ ΠΛΑΤΦΟΡΜΑΣ

### 2.1. Microsoft Visual Studio

Πρόκειται για ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) το οποίο χρησιμοποιείται για την ανάπτυξη desktop και διαδικτυακών εφαρμογών, διαδικτυακών υπηρεσιών (web services) και εφαρμογών για κινητά (mobile apps). Το Visual Studio υποστηρίζει 36 διαφορετικές γλώσσες προγραμματισμού και δίνει τη δυνατότητα στον code editor και τον debugger που διαθέτει να υποστηρίζουν σχεδόν κάθε γλώσσα προγραμματισμού. Οι βασικότερες γλώσσες που ενσωματώνει είναι μεταξύ άλλων οι C, C++, Visual Basic, C#, JavaScript, TypeScript, XML, HTML, CSS αλλά ακόμη και Python, Ruby και Node.js με την εγκατάσταση των αντίστοιχων plug-in.

Το σημαντικότερο ίσως πλεονέκτημα του Visual Studio εντοπίζεται στον code editor που υποστηρίζει τα εξής:

- **IntelliSense:** στοιχείο συμπλήρωσης κώδικα που επιταχύνει τη διαδικασία κωδικοποίησης εφαρμογών μειώνοντας τα τυπογραφικά και άλλα κοινά λάθη.
- **Code Refactoring:** η διαδικασία αποσαφήνισης και απλοποίησης του σχεδιασμού του υπάρχοντος κώδικα, χωρίς να αλλάζει η συμπεριφορά του.

Όσον αφορά τον ενσωματωμένο debugger, το Visual Studio έχει τη δυνατότητα να εντοπίζει σφάλματα τόσο σε επίπεδο πηγαίου κώδικα όσο και σε επίπεδο μηχανής ενώ ο designer του φιλοξενεί διάφορους visual designers για τη διευκόλυνση ανάπτυξης των εφαρμογών. [3]

Συνοπτικά πρόκειται για ένα πανίσχυρο εργαλείο για την ανάπτυξη κάθε είδους εφαρμογών για οποιαδήποτε πλατφόρμα, που παρέχει στον προγραμματιστή μία πληθώρα από υποστηριζόμενες γλώσσες προγραμματισμού αλλά και σημαντικά εργαλεία και δυνατότητες.

## 2.2. Η πλατφόρμα .NET

Η .NET είναι μία πλατφόρμα ανοικτού κώδικα που χρησιμοποιείται για την ανάπτυξη διαφόρων τύπων εφαρμογών. Επιτρέπει τη χρήση διαφόρων γλωσσών προγραμματισμού όπως η C#, η F# και η Visual Basic, επεξεργαστών κώδικα (editors) και βιβλιοθηκών για την ανάπτυξη διαδικτυακών και desktop εφαρμογών, mobile apps, εφαρμογών IoT, παιχνίδια κ.ά. Χαρακτηρίζεται ως cross-platform αφού δίνει τη δυνατότητα στον προγραμματιστή ανεξάρτητα από ποια γλώσσα χρησιμοποιεί να μπορεί να εκτελεί τον κώδικά του τοπικά σε οποιοδήποτε συμβατό λειτουργικό σύστημα. Οι παρακάτω εφαρμογές της .NET, επεκτείνουν τη συγκεκριμένη πλατφόρμα παρέχοντας εξειδικευμένη υποστήριξη για την ανάπτυξη των διάφορων τύπων εφαρμογών:

- **.NET Core:** Ανάπτυξη cross-platform ιστοσελίδων, διακομιστών και εφαρμογών κονσόλας σε περιβάλλον Windows, Linux, και macOS.
- **.NET Framework:** Ανάπτυξη ιστοσελίδων, υπηρεσιών, desktop εφαρμογών, κ.ά σε περιβάλλον Windows.
- **Xamarin:** Ανάπτυξη εφαρμογών για όλα τα μεγάλα λειτουργικά συστήματα των κινητών τηλεφώνων.

### 2.2.1. Η αρχιτεκτονική του .NET Framework

Τα δύο βασικά components του .NET Framework είναι το Common Language Runtime (CLR) και το .NET Framework Class Library.

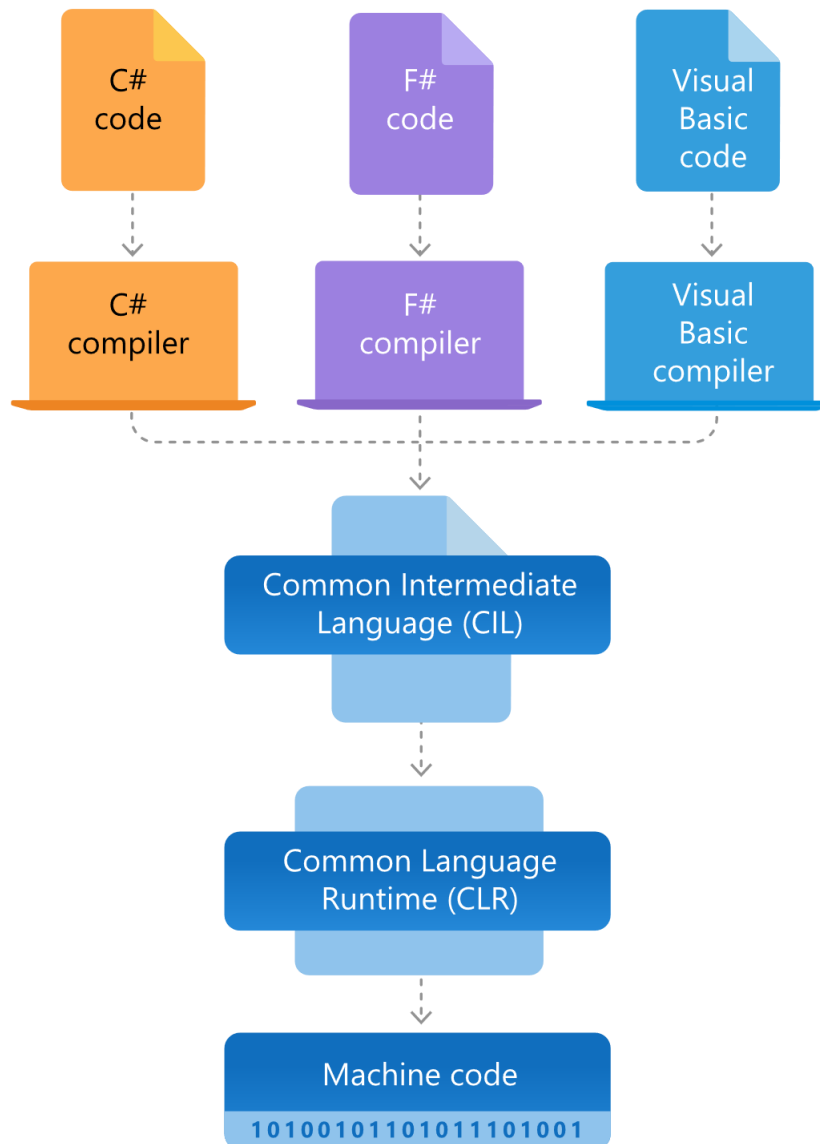
Το CLR είναι η μηχανή εκτέλεσης που χειρίζεται τις εφαρμογές που εκτελούνται. Οι υπηρεσίες που παρέχει αναφέρονται κυρίως στη διαχείριση νημάτων (thread management), τη διαχείριση της μνήμης (garbage collection), την ασφάλεια των τύπων δεδομένων (type-safety), τη διαχείριση εξαιρέσεων (exception handling) και πολλά άλλα.

Η Class Library περιλαμβάνει ένα σύνολο APIs και τύπων για κοινή λειτουργικότητα, δηλαδή επαναχρησιμοποιήσιμο κώδικα που οι προγραμματιστές μπορούν να ενσωματώσουν στις δικές τους εφαρμογές. Παρέχει τύπους για συμβολοσειρές, ημερομηνίες, αριθμούς κ.λπ.

καθώς και APIs για ανάγνωση και συγγραφή αρχείων, σύνδεση σε βάσεις δεδομένων, σχεδίαση και άλλα. [4]

Πέραν των δύο παραπάνω κύριων components του .NET Framework, το τελευταίο ενσωματώνει ακόμη μία πληθώρα σημαντικών υπηρεσιών όπως:

- **Διαχείριση μνήμης:** Στις εφαρμογές .NET Framework, το CLR παρέχει υπηρεσίες κατανομής και απελευθέρωσης μνήμης εκ μέρους της εφαρμογής ενώ είναι παράλληλα υπεύθυνο για το χειρισμό της διάρκειας ζωής των αντικειμένων.
- **Κοινό σύστημα τύπων:** Στο .NET Framework, οι βασικοί τύποι ορίζονται από το σύστημα τύπου και είναι κοινοί σε όλες τις γλώσσες που στοχεύουν το .NET Framework, τη στιγμή που στις παραδοσιακές γλώσσες προγραμματισμού, οι βασικοί τύποι ορίζονται από τον μεταγλωττιστή, ο οποίος περιπλέκει τη διαλειτουργικότητα μεταξύ γλωσσών.
- **Τεχνολογίες και πλαίσια ανάπτυξης εφαρμογών:** Το .NET Framework περιλαμβάνει βιβλιοθήκες για συγκεκριμένους τομείς ανάπτυξης εφαρμογών, όπως το ADO.NET για πρόσβαση δεδομένων, το ASP.NET για διαδικτυακές εφαρμογές, το WCF για εφαρμογές υπηρεσιών και το WPF για desktop εφαρμογές των Windows.
- **Γλωσσική διαλειτουργικότητα:** Με τη δυνατότητα που παρέχει η CIL όπως αυτή παράγεται από τους μεταγλωττιστές στο .NET Framework, οι ρουτίνες γραμμένες σε μία γλώσσα είναι προσβάσιμες και σε άλλες γλώσσες.
- **Συμβατότητα εκδόσεων:** Με σπάνιες εξαιρέσεις, οι εφαρμογές που αναπτύσσονται χρησιμοποιώντας μια συγκεκριμένη έκδοση του .NET Framework εκτελούνται χωρίς τροποποίηση σε μία οποιαδήποτε μεταγενέστερη έκδοση. [5]



Διάγραμμα 1: Η Αρχιτεκτονική του .NET Framework

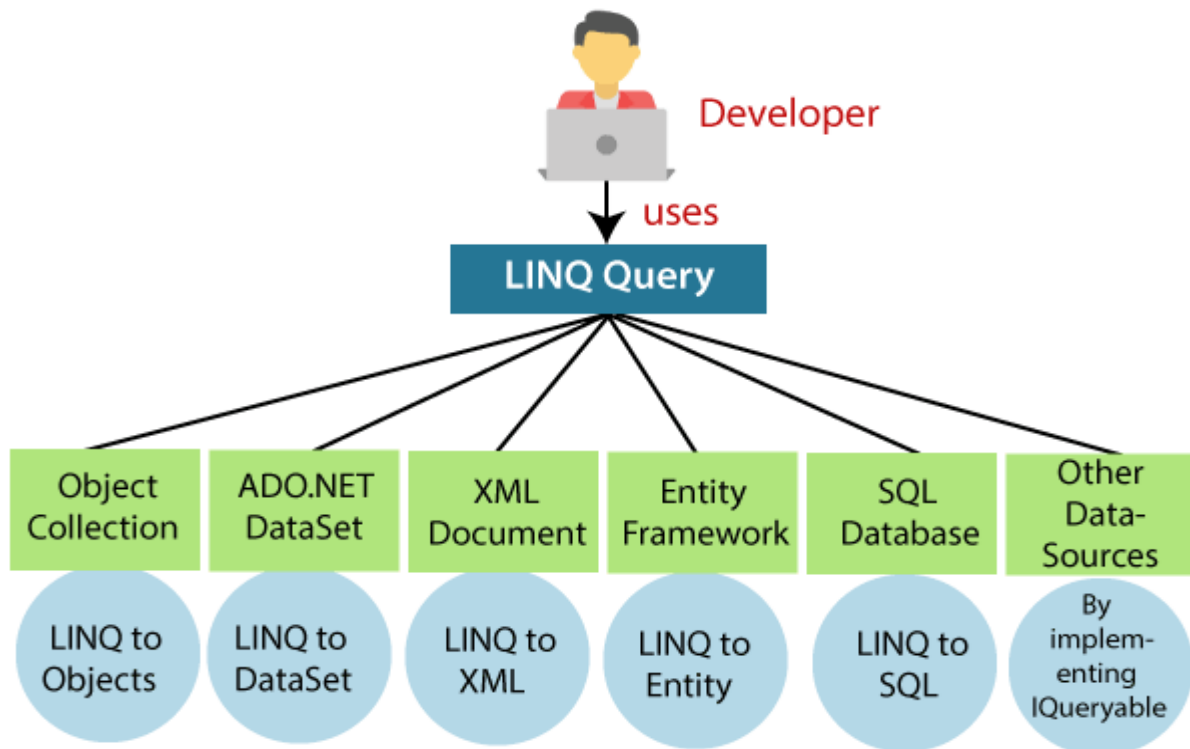
Στο .NET Framework ο κώδικας συντάσσεται σε ένα σύνολο δυαδικών εντολών, δηλαδή σε μια ενδιάμεση γλώσσα (CIL), όπως ορίζεται στο CLI. Οι εντολές της CIL εκτελούνται από ένα περιβάλλον εκτέλεσης συμβατό με το CLI, όπως το CLR. Θεωρητικά, αυτό εξαλείφει την ανάγκη διανομής διαφορετικών εκτελέσιμων αρχείων για διαφορετικές πλατφόρμες και τύπους επεξεργαστών. Ο μεταγλωττισμένος αυτός κώδικας αποθηκεύεται σε αρχεία με επέκταση .dll ή .exe.

Όταν εκτελείται μια εφαρμογή, το CLR διαβάζει το παραπάνω αρχείο και χρησιμοποιώντας έναν μεταγλωττιστή (JIT) το μετατρέπει σε κώδικα μηχανής που μπορεί να «τρέξει» στη συγκεκριμένη αρχιτεκτονική του υπολογιστή στον οποίο εκτελείται. [6]

#### 2.2.1.1. LINQ

Ένα κύριο συστατικό του .NET Framework είναι το LINQ. Πρόκειται για ένα σύνολο τεχνολογιών που βασίζονται στην ενσωμάτωση των δυνατοτήτων ερωτήματος (query) απευθείας στη γλώσσα C#. Παραδοσιακά, τα ερωτήματα σε δεδομένα εκφράζονται ως απλές συμβολοσειρές χωρίς να υποστηρίζεται ο έλεγχος τύπων κατά τη μεταγλώττιση ή η δυνατότητα IntelliSense. Επιπρόσθετα, για κάθε τύπο προέλευσης δεδομένων, όπως βάσεις δεδομένων SQL, έγγραφα XML και διάφορες διαδικτυακές υπηρεσίες, υπάρχουν διαφορετικές γλώσσες ερωτημάτων. Με τις τεχνολογίες του LINQ παρέχεται μία συνεπής εμπειρία ερωτημάτων για αντικείμενα (LINQ to Objects), σχεσιακές βάσεις δεδομένων (LINQ to SQL) και εγγράφων XML (LINQ to XML).

Με τη βοήθεια του LINQ, χρησιμοποιώντας σύνταξη ερωτήματος, δίνεται η δυνατότητα φιλτραρίσματος, ταξινόμησης και ομαδοποίησης των δεδομένων με ελάχιστο κώδικα. Με τη χρήση των ίδιων βασικών μοτίβων έκφρασης, υποβάλλονται ερωτήματα για την εξαγωγή, την επεξεργασία, τη μετατροπή, το μετασχηματισμό και το χειρισμό των δεδομένων σε βάσεις δεδομένων SQL, σε Datasets ADO.NET, σε έγγραφα XML, σε .NET συλλογές αντικειμένων αλλά ακόμη και στο Entity Framework. [7]



Διάγραμμα 2: Οι τεχνολογίες του LINQ

### 2.2.2. ASP.NET

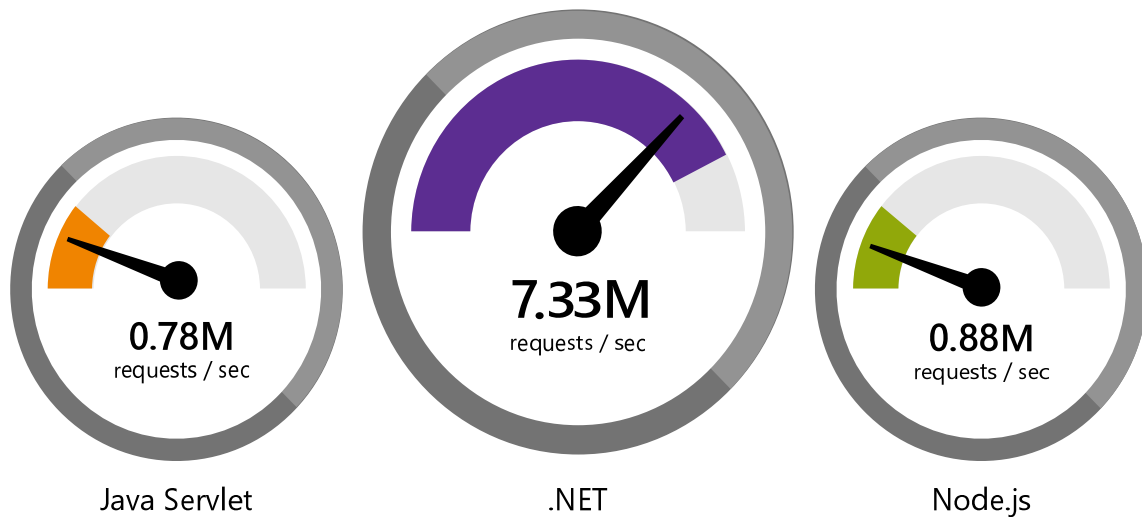
Το ASP.NET είναι ένα πρότυπο που επεκτείνει την πλατφόρμα .NET με εργαλεία και βιβλιοθήκες ειδικά για τη δημιουργία δυναμικών διαδικτυακών εφαρμογών. Μερικές προσθήκες του ASP.NET είναι μεταξύ άλλων ένα βασικό πλαίσιο για την επεξεργασία διαδικτυακών αιτημάτων, ένα πρότυπο σύνταξης δυναμικών ιστοσελίδων γνωστό ως RAZOR, βιβλιοθήκες για κοινά πρότυπα όπως το μοντέλο MVC, ένα σύστημα ελέγχου ταυτότητας που περιλαμβάνει βιβλιοθήκες, πρότυπες ιστοσελίδες, μια βάση δεδομένων για το χειρισμό των συνδέσεων και την ταυτοποίησή τους, αλλά και επεκτάσεις του code editor για τη διευκόλυνση της ανάπτυξης ιστοσελίδων. [8]

#### 2.2.2.1. ASP.NET Core

Το ASP.NET Core κυκλοφόρησε πρώτη φορά το 2016 και είναι μία εκδοχή του ASP.NET, ανοιχτού κώδικα που εκτελείται σε περιβάλλον Windows, Linux και macOS. Το



βασικότερο πλεονέκτημά του εντοπίζεται στις επιδόσεις του καθώς είναι σημαντικά ταχύτερο συγκριτικά με άλλα γνωστά πρότυπα ανάπτυξης διαδικτυακών εφαρμογών, όπως φαίνεται και στο παρακάτω διάγραμμα. [9]



Διάγραμμα 3: Απόδοση του ASP.NET Core

#### 2.2.2.2. ASP.NET Core MVC

Το ASP.NET Core MVC είναι ένα «ελαφρύ» πρότυπο, ανοιχτού κώδικα, βελτιστοποιημένο για χρήση με το ASP.NET Core, προσφέροντας τη δυνατότητα ανάπτυξης δυναμικών ιστοσελίδων με σαφή διαχωρισμό των εννοιών της εφαρμογής, γεγονός που προσφέρει σημαντικές δυνατότητες συντήρησης και ελέγχου. Η αρχιτεκτονική στην οποία είναι βασισμένο το πρότυπο MVC διαχωρίζει μία εφαρμογή σε 3 κύρια components, Models, Views και Controllers. Ειδικότερα:

- **Models:** Πρόκειται για components που δηλώνουν την κατάσταση της εφαρμογής, υλοποιούν το business logic της, και εκτελούν τις διαδικασίες που ορίζονται σε αυτήν. Είναι δηλαδή οι κύριες οντότητες της εφαρμογής, καθώς η συμπεριφορά της εφαρμογής προσδιορίζεται από την κατάσταση των αντικειμένων αυτών.

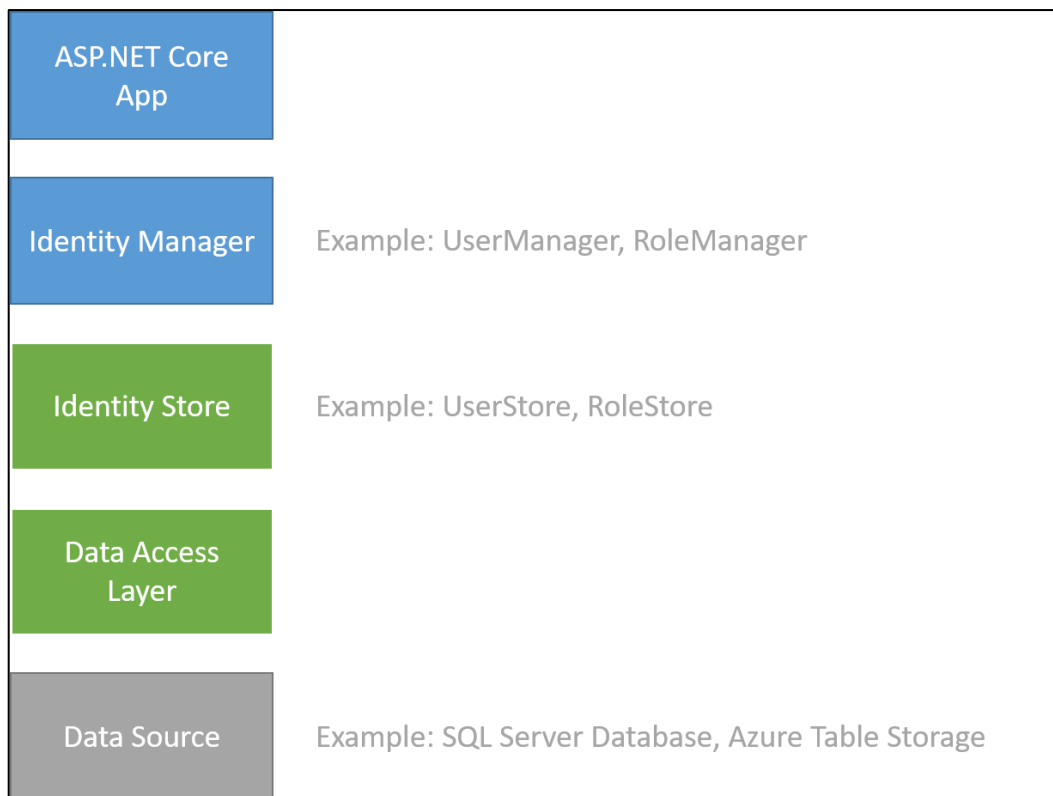
- **Views:** Είναι τα components που είναι υπεύθυνα για την παρουσίαση περιεχομένου μέσα από το UI της εφαρμογής. Χρησιμοποιούν το πρότυπο RAZOR για την ενσωμάτωση κώδικα .NET μέσα σε κώδικα HTML.
- **Controllers:** Είναι τα components που αναλαμβάνουν την αλληλεπίδραση με τον χρήστη. Πρόκειται για το αρχικό σημείο αναφοράς της εφαρμογής που λαμβάνει τα δεδομένα που εισάγει ο χρήστης, επιλέγει και επεξεργάζεται το μοντέλο και κάνει την επιλογή του κατάλληλου View με το οποίο θα παρουσιάσει τα αποτελέσματα στον χρήστη. Ο Controller δηλαδή χειρίζεται το τρόπο με τον οποίο η εφαρμογή θα ανταποκριθεί σε κάθε αίτημα του χρήστη.

Σε κάθε περίπτωση, χρησιμοποιώντας το πρότυπο MVC, τα αιτήματα του χρήστη δρομολογούνται σε έναν Controller, ο οποίος με τη σειρά του επιλέγει το κατάλληλο μοντέλο που θα επεξεργαστεί, ανακτά δεδομένα από τη βάση και εκτελεί τις εντολές του χρήστη. Στη συνέχεια ο Controller επιλέγει το επιθυμητό View με το οποίο θα προβάλει τα αποτελέσματα στο χρήστη, αφού ενσωματώσει τα όποια δεδομένα του μοντέλου που απαιτούνται. [10]

#### 2.2.2.3. ASP.NET Core Identity

Το ASP.NET Core περιλαμβάνει πλέον ένα σύστημα Identity, το οποίο επιτρέπει την προσθήκη της λειτουργίας login σε μια εφαρμογή, υλοποιώντας στην ουσία το Authentication τμήμα της ασφάλειας μιας εφαρμογής. Δίνεται η δυνατότητα δημιουργίας λογαριασμών από τους χρήστες, με όνομα χρήστη και κωδικό για τη σύνδεσή τους στην εφαρμογή, η οποία εναλλακτικά μπορεί να γίνει ακόμη και με εξωτερικούς παρόχους σύνδεσης όπως Facebook, Google, Twitter κ.ά.

Ουσιαστικά, το Identity είναι ένα API που διαχειρίζεται χρήστες, κωδικούς, ρόλους, αξιώσεις, tokens και πολλά άλλα δεδομένα που αφορούν την σύνδεση των χρηστών σε μια εφαρμογή. Τυπικά, το Identity διαμορφώνεται χρησιμοποιώντας μια βάση δεδομένων σε έναν SQL διακομιστή για την αποθήκευση των δεδομένων αυτών. [11] Το παρακάτω διάγραμμα παρουσιάζει τον τρόπο με τον οποίο διαρθρώνεται η αλληλεπίδραση μιας εφαρμογής με τη βάση δεδομένων στα πλαίσια υλοποίησης του Identity. Συγκεκριμένα, το Identity χρησιμοποιεί τους managers και τα stores για την διασύνδεση της εφαρμογής με το data access layer.



Διάγραμμα 4: Η αρχιτεκτονική του ASP.NET Core Identity

### 2.3. Η γλώσσα προγραμματισμού C#

Η C# παρουσιάστηκε από τη Microsoft στα τέλη του 2000 και υιοθετήθηκε ως de facto γλώσσα προγραμματισμού για την .NET πλατφόρμα. Είναι μία αντικειμενοστραφής γλώσσα προγραμματισμού, υψηλού επιπέδου, φιλική προς το χρήστη που διαθέτει έτοιμες κλάσεις και μία πληθώρα βιβλιοθηκών παρέχοντας έναν ενιαίο δομημένο τρόπο επικοινωνίας του χρήστη με το υλικό του υπολογιστικού συστήματος. Πρόκειται για μία γλώσσα που προέρχεται από την οικογένεια των γλωσσών προγραμματισμού C, προσφέροντας μία σχετική οικειότητα στους προγραμματιστές των γλωσσών C, C++, Java και JavaScript.

Η C# εκπληρώνει τις ανάγκες μιας γλώσσας εύκολης στη σύνταξη, την ανάγνωση και τη συντήρησή της συνδυάζοντας τις δυνατότητες και την ευελιξία της C++. Τα βασικά χαρακτηριστικά της είναι:

- **Απλή (Simple):** Η C# είναι μία γλώσσα που απλοποιεί την C++ εξαλείφοντας ενοχλητικούς τελεστές, παρέχει απλουστευμένο συντακτικό και οργάνωση στον

κώδικά της, μειώνοντας την πιθανότητα σφαλμάτων κατά τη σύνταξη, ενώ παράλληλα είναι εναρμονισμένη με σύγχρονες προγραμματιστικές πρακτικές.

- **Συνεπής (Consistent):** Η C# υποστηρίζει ένα ενοποιημένο σύστημα τύπων που εξαλείφει το πρόβλημα των διαφόρων κατηγοριών ακέραιων τύπων. Πλέον όλοι οι τύποι δεδομένων αντιμετωπίζονται ως αντικείμενα και οι προγραμματιστές έχουν τη δυνατότητα να επεκτείνουν το σύστημα τύπων απλά και εύκολα.
- **Σύγχρονη (Modern):** Ο λόγος που η C# καλείται σύγχρονη είναι το πλήθος των στοιχείων που υποστηρίζει, όπως η αυτόματη εκκαθάριση της μη χρησιμοποιούμενης μνήμης (garbage collection), η χρήση του δεκαδικού (decimal) τύπου δεδομένων για χρηματοοικονομικές εφαρμογές, το εξελιγμένο μοντέλο διαχείρισης λαθών (error handling), η μοντέρνα προσέγγιση του εντοπισμού σφαλμάτων (debugging) αλλά και το ισχυρό μοντέλο ασφάλειας.
- **Αντικειμενοστραφής (Object-Oriented):** Από τα θεμέλιά της η C# σχεδιάστηκε να είναι αντικειμενοστραφής. Υποστηρίζει και τα τρία δόγματα των αντικειμενοστραφών συστημάτων και συγκεκριμένα την ενθυλάκωση (encapsulation), την κληρονομικότητα (inheritance) και τον πολυμορφισμό (polymorphism).
- **Ασφαλής ως προς τους τύπους (Type-Safe):** Η C# ενσωματώνει ένα πλήθος από μέτρα που την καθιστά ασφαλή ως προς τους τύπους ευνοώντας την ανάπτυξη ισχυρών προγραμμάτων.
- **Δυνατότητα εκδόσεων (Versionable):** Δίνεται η δυνατότητα δημιουργίας νέων εκδόσεων ενοτήτων λογισμικού και η λειτουργία αυτών σε μία υπάρχουσα εφαρμογή (versioning). Μπορεί δηλαδή να αναβαθμιστεί ένα τμήμα του κώδικα σε μία εφαρμογή χωρίς να προκαλείται κάποια διένεξη με τον υπάρχοντα παλιό κώδικα.
- **Συμβατή (Compatible):** Οι προδιαγραφές της κοινής γλώσσας της .NET επιβάλλεται από την C#, γεγονός που επιτρέπει τη διαλειτουργία με άλλες γλώσσες.
- **Ευέλικτη (Flexible):** Η C# παρέχει την ικανότητα προσαρμογής των προγραμμάτων σε πιθανές ή μελλοντικές αλλαγές στις απαιτήσεις τους.

- **Διαλειτουργική (Inter-operable):** Υποστηρίζεται η χρήση COM αντικειμένων, χωρίς να λαμβάνεται υπόψη η γλώσσα προγραμματισμού που χρησιμοποιήθηκε για την σύνταξή τους. Επιπλέον, μπορεί να έχει απευθείας πρόσβαση σε components διαχειριζόμενου κώδικα στο CLR. [12]

## 2.4. HTML

Είναι η κύρια γλώσσα σήμανσης που ορίζει πώς οι ιστοσελίδες δημιουργούνται και παρουσιάζονται, καθώς τα στοιχεία της είναι τα βασικά δομικά στοιχεία των ιστοσελίδων. Με την χρήση των διαφόρων ετικετών HTML (tags) που διαθέτει η γλώσσα αυτή, δίνεται η δυνατότητα καθορισμού της δομής και των περιεχομένων μιας ιστοσελίδας (κείμενα, εικόνες, υπερσυνδέσεις κ.λπ.).

Ένα πρόγραμμα περιήγησης διαβάζει HTML κώδικα και βάσει αυτού μπορεί να συνθέσει μία ιστοσελίδα. Οι ετικέτες δεν εμφανίζονται αλλά χρησιμοποιούνται από το πρόγραμμα περιήγησης για την παρουσίαση του περιεχομένου της ιστοσελίδας. Σε γενικές γραμμές η HTML προσδιορίζει τη σημασία και τη δομή μιας ιστοσελίδας, τη στιγμή που άλλες τεχνολογίες όπως η CSS και η JavaScript χρησιμοποιούνται για να περιγράψουν την εμφάνιση και τη λειτουργικότητά της αντίστοιχα.

## 2.5. CSS

Η CSS είναι μια απλή γλώσσα που χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου που έχει γραφτεί με μια γλώσσα σήμανσης. Χρησιμοποιείται δηλαδή για την επεξεργασία της εμφάνισης εγγράφων HTML και XHTML, και κατά συνέπεια για τον έλεγχο της εμφάνισης μιας ιστοσελίδας και γενικότερα ενός ιστοτόπου. Η CSS είναι μια γλώσσα που διαμορφώνει στυλιστικά μία ιστοσελίδα παραμετροποιώντας χαρακτηριστικά, χρώματα, στοίχιση, δίνοντας παράλληλα περισσότερες δυνατότητες σε σχέση με την HTML.

Τα κυριότερα πλεονεκτήματα που παρέχει είναι:

- Ευκολία στη συντήρηση των ιστοσελίδων, καθώς η εμφάνιση τους μπορεί να ελέγχεται αποκλειστικά και μόνο από ένα αρχείο CSS, γεγονός που παρέχει τη δυνατότητα της εύκολης παραμετροποίησης σε περίπτωση αλλαγών.
- Ευελιξία στη μορφοποίηση των ιστοσελίδων που θα ήταν διαφορετικά δύσκολη ή και αδύνατη με τη χρήση της κλασσικής HTML.
- Σύμπτυξη των κανόνων μορφοποίησης σε ένα μοναδικό αρχείο, που μειώνει το μέγεθος των δεδομένων που θα πρέπει να φορτώσει το πρόγραμμα περιήγησης για να παρουσιάσει μία ιστοσελίδα.
- Δεδομένης της ικανότητας του προγράμματος περιήγησης να κατεβάζει στην cache του το αρχείο CSS ώστε να το χρησιμοποιεί κάθε φορά που απαιτείται για τη φόρτωση μίας σελίδας του ιστοτόπου και σε συνδυασμό με τον προηγούμενο κανόνα, μειώνεται και ο χρόνος φόρτωσης της σελίδας.

## 2.6. JavaScript

Πρόκειται για μια δυναμική, αντικειμενοστραφή, διερμηνευμένη γλώσσα προγραμματισμού υψηλού επιπέδου που δεν χρειάζεται μεταγλώττιση. Απέκτησε μεγάλη επιτυχία ως client-side γλώσσα για εκτέλεση κώδικα σε ιστοσελίδες και περιλήφθηκε σε διάφορα προγράμματα περιήγησης. Αυτό σημαίνει ότι η επεξεργασία του κώδικά της και η παραγωγή του τελικού περιεχομένου HTML δεν πραγματοποιείται στο διακομιστή (server), αλλά στο πρόγραμμα περιήγησης των χρηστών (client). Ενώ είναι κυρίως γνωστή ως scripting γλώσσα για ιστοσελίδες, ωστόσο χρησιμοποιείται σε πολλά περιβάλλοντα χωρίς την χρήση προγράμματος περιήγησης όπως στο Node.js ή το Apache CouchDB.

Η JavaScript ήρθε να αντιμετωπίσει το πρόβλημα της στασιμότητας της HTML, το περιεχόμενο της οποίας δεν μπορεί να μεταβληθεί δυναμικά. Για να μεταβληθεί το περιεχόμενό της απαιτούνταν η επικοινωνία με το διακομιστή για την επεξεργασία και αποστολή εκ νέου των δεδομένων στο πρόγραμμα περιήγησης του χρήστη, γεγονός που αύξανε δραματικά τον υπολογιστικό φόρτο των διακομιστών. Τη λύση στο πρόβλημα αυτό έδωσε η JavaScript η οποία εκτελείται στην πλευρά του χρήστη και δεν απαιτεί την επικοινωνία με τον διακομιστή για τη μεταβολή του περιεχομένου της ιστοσελίδας.

Συνοπτικά, μερικά πλεονεκτήματα που προσφέρει η JavaScript είναι τα εξής:

- Λιγότερη αλληλεπίδραση με το διακομιστή, αφού η επικύρωση των δεδομένων εισόδου των χρηστών μπορεί να πραγματοποιηθεί πριν την αποστολή τους στο διακομιστή με αποτέλεσμα η κίνηση δεδομένων μεταξύ χρήστη και διακομιστή να περιορίζεται, όπως και το υπολογιστικό φορτίο στον τελευταίο.
- Άμεση ανατροφοδότηση στους χρήστες, καθώς δεν απαιτείται η επικοινωνία με το διακομιστή και η επαναφόρτωση μιας σελίδας για την επικύρωση των δεδομένων.
- Αυξημένη διαδραστικότητα, με τη δημιουργία διεπαφών που αντιδρούν με συγκεκριμένες ενέργειες των χρηστών, όπως την αιώρηση με το ποντίκι (mouse hover) ή την ενεργοποίησή τους μέσω του πληκτρολογίου.
- Εξελιγμένες διεπαφές, όπως η κατασκευή drag-and-drop στοιχείων ή sliders με τη χρήση της JavaScript για τη δημιουργία ενός πλουσιότερου περιβάλλοντος για τους χρήστες μιας ιστοσελίδας.

Αν και η JavaScript αποτελεί μία από τις πιο ευρέως διαδεδομένες γλώσσες, η χρήση της υπόκειται σε περιορισμούς και δεν μπορεί να αντιμετωπιστεί ως μια πλήρης γλώσσα προγραμματισμού, αφού δε διαθέτει σημαντικά χαρακτηριστικά:

- Η JavaScript που εκτελείται στο client-side δεν επιτρέπει την ανάγνωση ή σύνθεση αρχείων για λόγους ασφαλείας.
- Δεν μπορεί να χρησιμοποιηθεί στις εφαρμογές δικτύωσης επειδή δεν υποστηρίζεται.
- Δεν έχει δυνατότητες πολλαπλών νημάτων ή επεξεργασιών (multi-threading/multi-processor). [13]

## 2.7. jQuery

Η jQuery είναι μία ανοιχτού κώδικα, μικρή, γρήγορη και πλούσια βιβλιοθήκη της JavaScript. Έχοντας αλλάξει τον τρόπο με τον οποίο εκατομμύρια άνθρωποι γράφουν

JavaScript, απλοποιεί τον χειρισμό και την διάτρεξη (επιλογή στοιχείων του DOM) HTML εγγράφων, την διαχείριση συμβάντων, την δημιουργία κίνησης στο CSS ενώ διευκολύνει και τη χρήση της Ajax, με ένα εύχρηστο API που λειτουργεί σε όλους σχεδόν τα γνωστά προγράμματα περιήγησης. [14]

## 2.8. Ajax

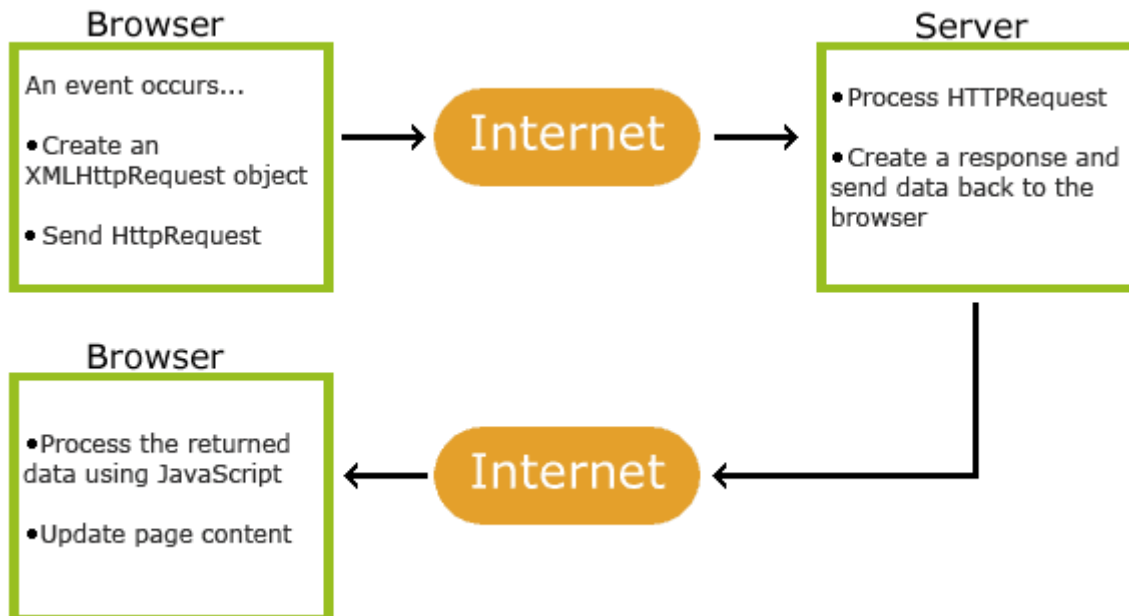
Παραδοσιακά, οι ιστοσελίδες χρειαζόντουσαν επαναφόρτωση για να ενημερώσουν το περιεχόμενό τους. Για παράδειγμα στις ιστοσελίδες ηλεκτρονικού ταχυδρομείου αυτό σήμαινε ότι οι χρήστες έπρεπε να φορτώσουν ξανά τη σελίδα των εισερχομένων τους με μη αυτόματο τρόπο για να ελέγξουν και να δουν αν είχαν νέα μηνύματα. Αυτό είχε ως αποτέλεσμα να αποτελεί μία αρκετά αργή διαδικασία που απαιτούσε την είσοδο του χρήστη. Όταν ο χρήστης ξαναφόρτωνε τα εισερχόμενά του, ο διακομιστής έπρεπε να ανακατασκευάσει ολόκληρη την ιστοσελίδα και να ξαναστείλει ολόκληρο το HTML, CSS, JavaScript, καθώς και το email του χρήστη, με εξαιρετικά αναποτελεσματικό και αποδοτικό τρόπο. Στην ιδανική περίπτωση, ο διακομιστής θα πρέπει να στέλνει μόνο τα νέα μηνύματα του χρήστη και όχι ολόκληρη τη σελίδα.

Μέχρι το 2003, όλα τα μεγάλα προγράμματα περιήγησης έλυσαν αυτό το ζήτημα υιοθετώντας το αντικείμενο XMLHttpRequest (XHR), δίνοντας στα προγράμματα περιήγησης τη δυνατότητα να επικοινωνούν με τον διακομιστή χωρίς να απαιτείται επαναφόρτωση της σελίδας. Το XHR είναι μέρος μιας τεχνολογίας, ονόματι Ajax. Η Ajax δεν είναι μία γλώσσα προγραμματισμού, αλλά μία τεχνολογία που εκτελείται στο client-side των προγραμμάτων περιήγησης με την οποία οι διαδικτυακές εφαρμογές μπορούν να στέλνουν και να ανακτούν δεδομένα από έναν διακομιστή ασύγχρονα, χωρίς να επεμβαίνουν στη συμπεριφορά και την εμφάνιση της υπάρχουσας σελίδας. Χρησιμοποιώντας το Ajax, τα δεδομένα μπορούν να μεταφερθούν μεταξύ του προγράμματος περιήγησης και του διακομιστή, χρησιμοποιώντας το XHR API, χωρίς να χρειάζεται να γίνει επαναφόρτωση της ιστοσελίδας.

Τα αιτήματα Ajax ενεργοποιούνται από κώδικα JavaScript. Ειδικότερα, αποστέλλεται ένα αίτημα σε μια διεύθυνση URL και όταν επιστρέφει μια απάντηση, μπορεί να ενεργοποιηθεί μια λειτουργία επανάκλησης για να χειριστεί την απόκριση. Επειδή το αίτημα είναι ασύγχρονο, ο υπόλοιπος κώδικας της ιστοσελίδας συνεχίζει να εκτελείται κατά τη διεκπεραίωση του



αιτήματος, επομένως είναι επιτακτική ανάγκη να χρησιμοποιηθεί μια επιστροφή κλήσης για τη διαχείριση της απόκρισης.



Διάγραμμα 5: Η λειτουργία του Ajax

Δυστυχώς, τα διάφορα προγράμματα περιήγησης εφαρμόζουν το Ajax API με διαφορετικό τρόπο, κάτι που σημαίνει ότι οι προγραμματιστές θα έπρεπε να μεριμνούν για όλα τα διαφορετικά προγράμματα περιήγησης για να διασφαλίσουν ότι το Ajax θα λειτουργούσε καθολικά. Ευτυχώς, το jQuery παρέχει υποστήριξη Ajax που αφαιρεί τις όποιες διαφορές του προγράμματος περιήγησης. Ωστόσο, οι περισσότερες εφαρμογές jQuery που υποστηρίζουν Ajax δεν χρησιμοποιούν στην πραγματικότητα XML, αλλά μεταφέρουν τα δεδομένα ως απλό HTML ή JSON. [15]

## 2.9. MySQL

Πρόκειται για ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS), ανοιχτού κώδικα που αναπτύχθηκε και υποστηρίζεται από την Oracle. Οι βάσεις δεδομένων της MySQL δίνουν δυνατότητες προσθήκης, πρόσβασης και επεξεργασίας στα αποθηκευμένα

δεδομένα. Οι βάσεις αυτές είναι σχεσιακές, που σημαίνει ότι τα δεδομένα είναι αποθηκευμένα σε ξεχωριστούς πίνακες στους οποίους οι διάφοροι τύποι δεδομένων μπορεί να σχετίζονται μεταξύ τους με σχέσεις που βοηθούν στη δόμηση των δεδομένων.

Η SQL από την πλευρά της είναι μία γλώσσα ερωτημάτων που χρησιμοποιείται από τους προγραμματιστές για τη δημιουργία, την επεξεργασία και την εξαγωγή δεδομένων από μία σχεσιακή βάση, ενώ είναι παράλληλα επιφορτισμένη με τον έλεγχο της πρόσβασης των χρηστών σε αυτήν. Έτσι με τη χρήση της MySQL υπάρχει η δυνατότητα υλοποίησης μίας σχεσιακής βάσης δεδομένων, γρήγορα, εύκολα και αξιόπιστα που να μπορεί να διαχειριστεί με τη βοήθεια της SQL. [16]

## 2.10. Entity Framework

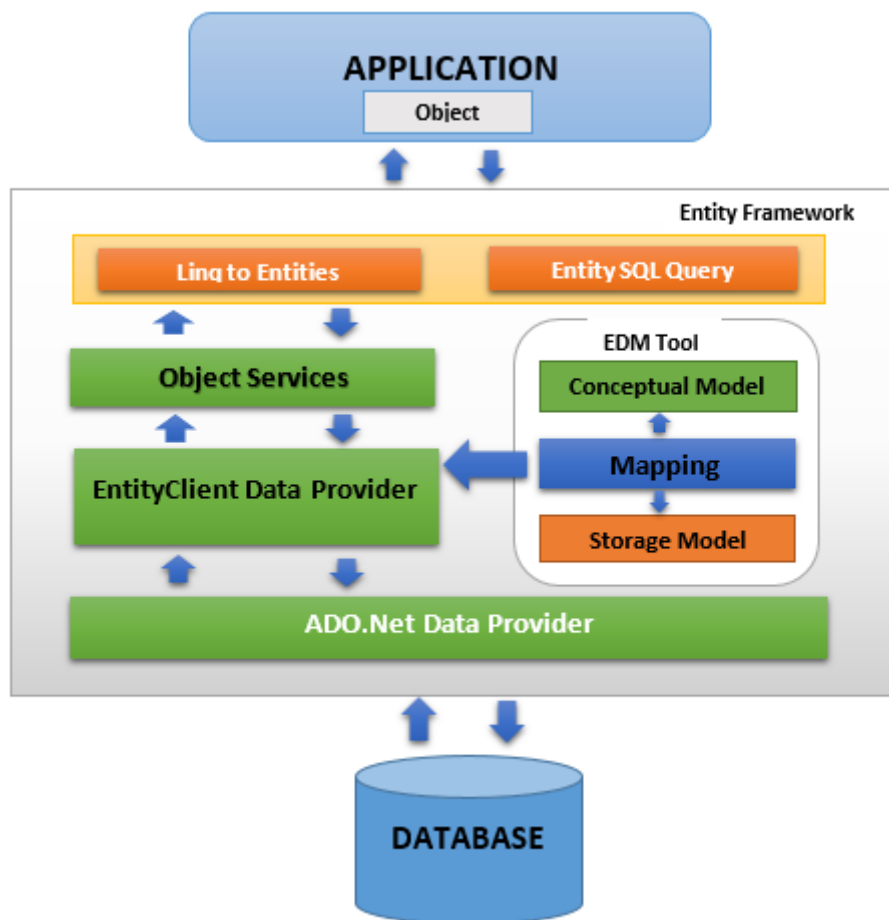
Πριν από την έκδοση .NET 3.5 οι προγραμματιστές χρησιμοποιούσαν επιπλέον κώδικα με αρκετές εντολές για την ανάκτηση και την εγγραφή δεδομένων σε μια βάση μιας εφαρμογής. Αυτή την επίπονη και επιρρεπή σε σφάλματα διαδικασία ήρθε να αντικαταστήσει η Microsoft με το EF, ένα πλαίσιο που αυτοματοποιεί όλες τις σχετικές διαδικασίες μεταξύ της εφαρμογής και της βάσης δεδομένων, όπως αναλύεται στο Διάγραμμα 6.

Το EF είναι ένα ORM framework ανοιχτού κώδικα, σχεδιασμένο για εφαρμογές .NET, που δίνει τη δυνατότητα στους προγραμματιστές να μεταχειρίζονται τα δεδομένα χρησιμοποιώντας αντικείμενα συγκεκριμένων κλάσεων χωρίς να απαιτείται η γνώση των επιμέρους στοιχείων της βάσης δεδομένων. Με τον τρόπο αυτό καθίσταται ευκολότερη η διαχείριση των δεδομένων, καθώς απαιτείται λιγότερος κώδικας για την πρόσβαση στα δεδομένα της βάσης. [17]

Η υλοποίηση του EF μπορεί να πραγματοποιηθεί με τρεις διαφορετικούς τρόπους:

- **Database first:** Σε αυτή την περίπτωση σχεδιάζονται οι πίνακες της βάσης και στη συνέχεια το EF αναλαμβάνει τη δημιουργία μιας domain class. Πρόκειται για την συνηθέστερη προσέγγιση που επιλέγουν οι προγραμματιστές για την ανάπτυξη των εφαρμογών τους.

- **Model first:** Με τη χρήση του Visual Designer που παρέχει το Visual Studio μοντελοποιούνται οι κλάσεις και οι συσχετίσεις τους, βάσει του οποίου το EF παράγει μια domain class καθώς και την βάση δεδομένων.
- **Code first:** Με την προσέγγιση αυτή ο προγραμματιστής αναλαμβάνει τη δημιουργία της domain class και το EF δημιουργεί στη συνέχεια την βάση δεδομένων. [18]



Διάγραμμα 6: Η αρχιτεκτονική του Entity Framework

### 3. ΥΛΟΠΟΙΗΣΗ ΤΗΣ ΠΛΑΤΦΟΡΜΑΣ

Σε αυτό το κεφάλαιο γίνεται ανάλυση των σταδίων υλοποίησης της πλατφόρμας καταχώρησης δράσεων εξοικονόμησης ενέργειας. Παρουσιάζεται η κύρια δομή της, καθώς και ορισμένες από τις βασικές λειτουργίες της. Παράλληλα, περιγράφεται ο τρόπος χειρισμού των προαναφερθέντων εργαλείων και τεχνολογιών που χρησιμοποιήθηκαν για την ανάπτυξή της. Σε αρχικό στάδιο δημιουργήθηκε μία βάση δεδομένων σε MySQL για την υποστήριξη της εφαρμογής, ενώ στη συνέχεια με τη χρήση του Visual Studio αναπτύχθηκε τόσο το UI συνδυάζοντας τεχνολογίες όπως HTML, CSS, JavaScript κ.ά, όσο και το λειτουργικό μέρος της με τη χρήση του EF, του ASP.NET Core και του μοντέλου MVC.

#### 3.1. Υλοποίηση της βάσης δεδομένων

Για την υποστήριξη της εφαρμογής απαιτήθηκε μία βάση δεδομένων στην οποία θα αποθηκεύονται οι χρήστες που θα κάνουν εγγραφή στην πλατφόρμα καθώς και τα στοιχεία τους, οι δράσεις εξοικονόμησης ενέργειας που θα καταχωρούνται από τις επιχειρήσεις αλλά και οι επενδύσεις που θα αναλαμβάνουν οι επενδυτές για την υλοποίηση αυτών των δράσεων. Για τη δημιουργία της βάσης χρησιμοποιήθηκε το MySQL Workbench ενώ για τη λειτουργία του Server που θα υποστηρίξει τη βάση μας επιλέχθηκε το εργαλείο XAMPP. Σε κάθε πίνακα υπάρχει ένα πεδίο Id το οποίο χρησιμοποιείται για την ταυτοποίηση της κάθε εγγραφής (PK) αλλά και για τη διασύνδεση των πινάκων με τη δημιουργία FKs και είναι Auto Increment για την ελαχιστοποίηση του κώδικα που απαιτείται κατά την εισαγωγή νέων εγγραφών και την αποφυγή λαθών. Με τη διασύνδεση της εφαρμογής με τη βάση δεδομένων, όπως αναλύεται μετέπειτα, θα επιτευχθεί η πλήρης λειτουργικότητά της.

##### 3.1.1. Δημιουργία των πινάκων

Αρχικά, δημιουργήθηκε το DB Schema **energy** και σε πρώτο στάδιο απαιτήθηκε η δημιουργία των βοηθητικών πινάκων που θα χρησιμοποιηθούν. Συγκεκριμένα,

κατασκευάστηκε ο πίνακας **sector** που περιλαμβάνει τους τομείς στους οποίους δραστηριοποιούνται οι επιχειρήσεις και ο πίνακας **country** ο οποίος περιέχει τις χώρες της ΕΕ-27 και οι οποίες θα χρησιμοποιηθούν στην καταχώρηση της χώρας στην οποία κατοικούν οι επενδυτές, έχουν την ιθαγένειά τους και της χώρας στην οποία εδρεύουν οι επιχειρήσεις. Οι πίνακες **sector** και **country** συνδέονται ο καθένας με τους πίνακες **business** και **investor** με σχέση One to Many. Ο πίνακας **category** παραθέτει όλες τις κατηγορίες δράσεων εξοικονόμησης ενέργειας που θα χρησιμοποιηθούν στην εφαρμογή και ο πίνακας **status** την κατάσταση στην οποία βρίσκεται μία δράση, δηλαδή αν είναι σε αναμονή, σε εξέλιξη, αν έχει ολοκληρωθεί ή ακυρωθεί. Οι δύο αυτοί πίνακες συνδέονται με τον πίνακα **action**, καθένας με σχέση One to Many. Ο πίνακας **paymenttype** παρουσιάζει το είδος της ταμειακής ροής ανάλογα με το αν αφορά κεφάλαιο ή τόκο και συνδέεται με τον πίνακα **payments** με σχέση One to Many.

Έπειτα, δημιουργήθηκε ο πίνακας των χρηστών που θα κάνουν login στην εφαρμογή και με βάση τον οποίο θα επιτυγχάνεται η λειτουργία του ASP.NET Core Identity. Σε αυτό το σημείο διευκρινίζεται ότι ο εν λόγω πίνακας **user**, όπως εμφανίζεται στην Εικόνα 1, περιέχει τα στοιχεία του Username, Email και PasswordHash<sup>1</sup> τα οποία αφενός είναι κοινά και για τα δύο είδη χρηστών και αφετέρου είναι απαραίτητα για την εφαρμογή του Identity. Δεδομένου ότι ο πίνακας αυτός περιέχει μόνο τα απολύτως απαραίτητα στοιχεία για το Identity, γίνεται αντιληπτό ότι επιτυγχάνει και η βελτιστοποίηση της ταχύτητας εκτέλεσής του. Αν και το πεδίο Email μπορεί να χρησιμοποιηθεί για την είσοδο των χρηστών (πέραν του Username) ή ακόμη και για την αποτροπή δημιουργίας λογαριασμών με το ίδιο Email, ωστόσο δεν υλοποιείται αλλά δύναται να χρησιμοποιηθεί σε κάποια πιθανή επέκταση της εφαρμογής.

---

<sup>1</sup> Το Password αποθηκεύεται hashed στην βάση δεδομένων ώστε να επιτυγχάνεται η ασφάλεια της εφαρμογής και η διασφάλιση των δεδομένων των χρηστών.

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
Id	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Username	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Email	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
PasswordHash	VARCHAR(450)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Εικόνα 1: Πίνακας user

Στη συνέχεια υλοποιήθηκε η δημιουργία των πινάκων **business** και **investor** οι οποίοι θα χρησιμοποιηθούν για την αποθήκευση των στοιχείων των χρηστών ανάλογα με το είδος της εγγραφής. Οι δύο αυτοί πίνακες περιλαμβάνουν όλα τα απαραίτητα πεδία για κάθε κατηγορία χρήστη, όπως φαίνεται στις Εικόνες 2 και 3. Συνδέονται μέσω των FKs UserId με το PK (Id) του πίνακα user, καθώς και μέσω των FKs CountryId με το PK του πίνακα country. Επιπλέον, ο πίνακας business συνδέεται με το FK SectorId με το PK του πίνακα sector, ενώ ο πίνακας investor με το FK CitizenId με το PK του πίνακα country.

Ειδικότερα, ο πίνακας business περιλαμβάνει τα πεδία UserId, Name, SectorId, Url (το site της εταιρίας εφόσον υφίσταται), Street, StreetNumber, PostCode, City, CountryId, Tax (ΑΦΜ), Iban, ReprFName (όνομα αντιπροσώπου), ReprLname (επώνυμο αντιπροσώπου) και Phone, όπως φαίνεται στην Εικόνα 2.

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
Id	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
UserId	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Name	VARCHAR(100)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
SectorId	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Url	VARCHAR(100)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
Street	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
StreetNumber	VARCHAR(10)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
PostCode	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
City	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
CountryId	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Tax	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Iban	VARCHAR(30)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
ReprFName	VARCHAR(30)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
ReprLName	VARCHAR(30)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Phone	VARCHAR(14)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Εικόνα 2: Πίνακας business

Αντίστοιχα, ο πίνακας investor περιλαμβάνει τα πεδία UserId, Fname (Όνομα), Lname (Επώνυμο), Street, StreetNumber, PostCode, City, Phone, CountryId, CitizenId και Tax (ΑΦΜ) όπως φαίνεται στην Εικόνα 3.

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
Id	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
UserId	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Fname	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Lname	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Street	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
StreetNumber	VARCHAR(10)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
PostCode	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
City	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Phone	VARCHAR(14)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
CountryId	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
CitizenId	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Tax	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Εικόνα 3: Πίνακας investor

Κατόπιν, σχεδιάστηκε ο πίνακας **action**, ο οποίος θα χρησιμεύσει για την αποθήκευση των στοιχείων των δράσεων εξοικονόμησης ενέργειας που θα καταχωρούνται από τις επιχειρήσεις. Ο πίνακας αυτός, όπως φαίνεται στην Εικόνα 4, περιλαμβάνει τα πεδία Title (τίτλος δράσης), Description (περιγραφή), Investment (ύψος επένδυσης), StartDate (ημ/νία έναρξης), Duration (διάρκεια σε έτη), InterestRate (επιτόκιο), DiscountRate (προεξοφλητικό επιτόκιο), EnergySavings (εξοικονόμηση ενέργειας σε GJ), CostSavings (εξοικονόμηση σε €), EmissionsReduction (μείωση ρύπων σε tCO<sub>2</sub>e). Περιλαμβάνει ακόμη τα πεδία StatusId, CategoryId και BusinessId με τα οποία ως FKs συνδέεται με τους πίνακες category, status και business αντίστοιχα, ενώ η σχέση του με τον τελευταίο είναι Many to One.

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
Id	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
BusinessId	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
CategoryId	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
StatusId	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1
Title	VARCHAR(100)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Description	VARCHAR(1000)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Investment	DECIMAL(15,2)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
StartDate	DATE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Duration	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
InterestRate	DECIMAL(5,2)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
DiscountRate	DECIMAL(5,2)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
EnergySavings	DECIMAL(15,2)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
CostSavings	DECIMAL(15,2)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
EmissionsReduction	DECIMAL(15,2)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Εικόνα 4: Πίνακας action

Στη συνέχεια δημιουργήθηκε ο πίνακας actionsegment, ο οποίος θα περιέχει τις συμμετοχές των επενδυτών στις δράσεις εξοικονόμησης ενέργειας, που θα καταχωρούν οι επιχειρήσεις. Ο πίνακας αυτός, όπως φαίνεται και στην Εικόνα 5, περιλαμβάνει το πεδίο Amount (ποσό συμμετοχής), καθώς και τα FKs ActionId και InvestorId με τα οποία συνδέεται με τους πίνακες action και investor με σχέση Many to One με τον καθένα.

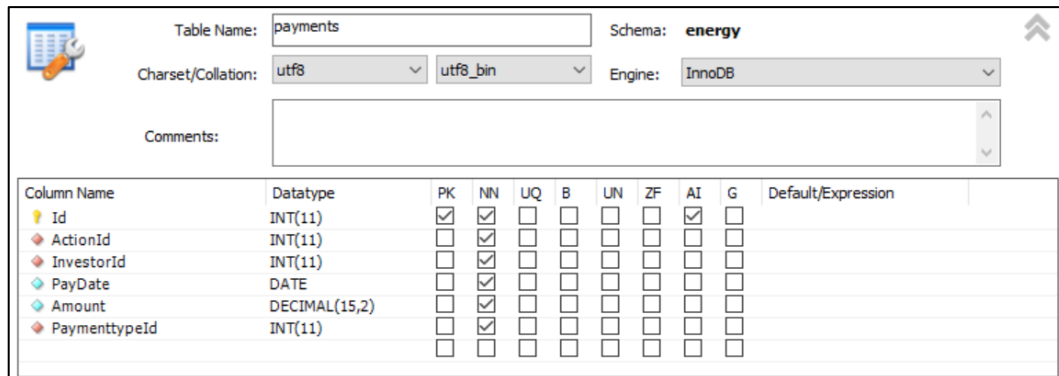
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
Id	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
ActionId	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
InvestorId	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Amount	DECIMAL(15,2)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Εικόνα 5: Πίνακας actionsegment

Τέλος, για την αποθήκευση των δεδομένων που αφορούν τις ταμειακές ροές από και προς τις επιχειρήσεις για τη χρηματοδότηση των δράσεων τόσο για το κεφάλαιο όσο και για τους τόκους, δημιουργήθηκε ο πίνακας payments, όπως Εικόνα 6. Περιλαμβάνει τα πεδία ActionId, InvestorId, PayDate (ημ/νία πληρωμής), Amount (ποσό πληρωμής) και PaymenttypeId. Τα FKs ActionId, InvestorId και PaymenttypeId χρησιμοποιούνται για τη



σύνδεση με τους πίνακες action, investor και paymenttype, ενώ η σχέση του με τους πίνακες action και investor είναι Many to One.



Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
Id	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
ActionId	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
InvestorId	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
PayDate	DATE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Amount	DECIMAL(15,2)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
PaymenttypeId	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Εικόνα 6: Πίνακας payments

### 3.1.2. Δημιουργία του View npv

Για τον υπολογισμό της Καθαρής Παρούσας Αξίας (ΚΠΑ - NPV) των δράσεων δημιουργήθηκε το View **npv**. Το View αυτό συνδυάζει στοιχεία από τους πίνακες payments και action για την παραγωγή των παρούσων αξιών των πληρωμών του πίνακα payments, θεωρώντας ως χρονική στιγμή  $T_0$  την ημερομηνία έναρξης της δράσης. Για κάθε χρονική στιγμή υπολογίζονται οι παρούσες αξίες βάσει του προεξοφλητικού επιτοκίου (DiscountRate), ενώ η εξοικονόμηση (CostSavings) που επιτυγχάνεται μέσω της υλοποίησης της δράσης, επιμερίζεται ισόποσα σε κάθε έτος ανάλογα με τη διάρκεια της δράσης. Το npv View περιλαμβάνει τα πεδία ActionId (το Id της δράσης), PayDate (ημ/νία πληρωμής), Investment (εισροή και επιστροφή του κεφαλαίου), Interests (πληρωμή τόκων), Savings (έσοδα από εξοικονόμηση) και NetValue (το άθροισμα των πεδίων Investments, Interests και Savings). Η δημιουργία του View npv υλοποιήθηκε με το παρακάτω script. Ως ΚΠΑ μιας δράσης νοείται το άθροισμα των πεδίων NetValue του npv με ActionId, το Id της δράσης στην οποία αναφέρεται.

```
01. CREATE
02.     ALGORITHM = UNDEFINED
03.     DEFINER = `root`@`localhost`
04.     SQL SECURITY DEFINER
05.     VIEW `npv` AS
06.     SELECT
07.         `c`.`ActionId` AS `ActionId`,
08.         `c`.`PayDate` AS `PayDate`,
09.         `c`.`Investment` AS `Investment`,
10.         `c`.`Interest` AS `Interest`,
11.         `c`.`Savings` AS `Savings`,
12.         ROUND((`c`.`Investment` + `c`.`Interest` + `c`.`Savings`) / POW(1 + `c`.`DiscountRate` / 100,
13.             EXTRACT(YEAR FROM `c`.`PayDate`) - EXTRACT(YEAR FROM `c`.`StartDate`))),
14.         2) AS `NetValue`
15.     FROM
16.     (SELECT
17.         `b`.`Id` AS `ActionId`,
18.         `a`.`PayDate` AS `PayDate`,
19.         SUM(IF(`a`.`PaymentTypeId` = 1, `a`.`Amount`, 0)) AS `Investment`,
20.         SUM(IF(`a`.`PaymentTypeId` = 2, `a`.`Amount`, 0)) AS `Interest`,
21.         0 AS `Savings`,
22.         `b`.`DiscountRate` AS `DiscountRate`,
23.         `b`.`StartDate` AS `StartDate`
24.     FROM
25.     (`energy`.`payments` `a`
26.     JOIN `energy`.`action` `b`)
27.     WHERE
28.         `a`.`ActionId` = `b`.`Id`
29.         AND `a`.`Amount` > 0
30.     GROUP BY `b`.`Id`, `a`.`PayDate`, `b`.`DiscountRate`, `b`.`StartDate`
31.     UNION
32.     SELECT
33.         `b`.`Id` AS `ActionId`,
34.         `a`.`PayDate` AS `PayDate`,
35.         SUM(IF(`a`.`PaymentTypeId` = 1, `a`.`Amount`, 0)) AS `sum(if(PaymentTypeId=1,amount,0))`,
36.         SUM(IF(`a`.`PaymentTypeId` = 2, `a`.`Amount`, 0)) AS `sum(if(PaymentTypeId=2,amount,0))`,
37.         ROUND(`b`.`CostSavings` / `b`.`Duration`, 2) AS `Round(CostSavings/Duration,2)`,
38.         `b`.`DiscountRate` AS `DiscountRate`,
39.         `b`.`StartDate` AS `StartDate`
40.     FROM
41.     (`energy`.`payments` `a`
42.     JOIN `energy`.`action` `b`)
43.     WHERE
44.         `a`.`ActionId` = `b`.`Id`
45.         AND `a`.`Amount` < 0
46.     GROUP BY `b`.`Id`, `a`.`PayDate`, `b`.`DiscountRate`, `b`.`StartDate`) `c`
47.     ORDER BY `c`.`ActionId`, `c`.`PayDate`
```

Script 1: Δημιουργία npv View

### 3.1.3. Υλοποίηση αυτόματης εισαγωγής δεδομένων στον πίνακα payments

Για την εισαγωγή των στοιχείων που αφορούν τις πληρωμές χρησιμοποιήθηκε μία Procedure που θα εισάγει τα στοιχεία στον πίνακα payments και μέσω ενός Trigger υλοποιείται η αυτοματοποίηση της διαδικασίας.

Ειδικότερα, δημιουργήθηκε το Trigger **actionsegment\_AFTER\_INSERT** στον πίνακα actionsegment, όπου κάθε φορά που ένας επενδυτής δημιουργεί μία καταχώρηση στον εν λόγω πίνακα, επιλέγοντας να επενδύσει σε μία δράση θα ενεργοποιεί το Trigger. Το τελευταίο αναλαμβάνει, όπως φαίνεται στο Script 2, να ελέγξει αν μετά την τελευταία καταχώρηση συγκεντρώθηκε το απαραίτητο ποσό που απαιτεί η υπόψιν δράση και στη συνέχεια εκτελεί την Procedure create\_payment\_periods.

```
01. CREATE DEFINER=`root`@`localhost` TRIGGER actionsegment_AFTER_INSERT AFTER INSERT ON actionsegment
02. FOR EACH ROW
03. BEGIN
04. Declare pos1,pos2 decimal(15,2);
05. Declare cnt int;
06.
07. select count(*) into cnt from payments where ActionId=New.ActionId;
08. if cnt=0 then
09.     select Investment into pos1 from action where Id=New.ActionId;
10.     select sum(Amount) into pos2 from actionsegment where ActionId=New.ActionId;
11.     if pos2>=pos1 then
12.         call create_payment_periods(New.ActionId);
13.     end if;
14. end if;
15. END
```

### Script 2: Δημιουργία Trigger actionsegment\_AFTER\_INSERT

Η Procedure **create\_payment\_periods** με τη σειρά της εισάγει τα δεδομένα στον πίνακα payments, βάσει του Script 3, ανάλογα με τη διάρκεια της δράσης, το επιτόκιο και για κάθε έναν επενδυτή αναλόγως του ποσού συμμετοχής του. Μετά την ολοκλήρωση της εισαγωγής των παραπάνω στοιχείων μεταβάλλει την κατάσταση της δράσης από «Pending» σε «In Progress».

```
01. CREATE DEFINER=`root`@`localhost` PROCEDURE `create_payment_periods`(in actId int)
02. BEGIN
03. DECLARE invstr, years INT;
04. DECLARE counter,done INT default 0;
05. DECLARE pos,interest decimal(15,2);
06. DECLARE sdate date;
07. DECLARE cur CURSOR FOR SELECT InvestorId,sum(Amount) FROM actionsegment WHERE ActionId=actId
08. GROUP BY InvestorId;
09. DECLARE CONTINUE HANDLER FOR NOT FOUND SET DONE=1;
10.
11. SELECT StartDate, Duration,InterestRate INTO sdate, years, interest FROM action WHERE Id=actId;
12.
13. open cur;
14. segm: LOOP
15. FETCH cur INTO invstr, pos;
16. IF done=1 THEN
17.     UPDATE action SET StatusId=2 WHERE Id=actId;
18.     LEAVE segm;
19. END IF;
20. while counter <= years do
21. if (counter=0) then
22.     insert into payments (ActionId,InvestorId,PaymenttypeId,PayDate,Amount)
23.     values (actId,invstr,1,sdate,pos);
24. else
25.     insert into payments (ActionId,InvestorId,PaymenttypeId,PayDate,Amount)
26.     values (actId,invstr,2,adddate(sdate,INTERVAL counter year),round(-pos*interest/100,2));
27. end if;
28. if (counter=years) then
29.     insert into payments (ActionId,InvestorId,PaymenttypeId,PayDate,Amount)
30.     values (actId,invstr,1,adddate(sdate,INTERVAL counter year),-pos);
31. end if;
32.     set counter=counter+1;
33. end while;
34. end loop segm;
35. close cur;
36. END
```

### Script 3: Δημιουργία Procedure create\_payment\_periods

#### 3.1.4. Υλοποίηση αυτόματης πρόκρισης της κατάστασης των δράσεων

Η κατάσταση στην οποία μπορεί να βρίσκεται μία δράση μπορεί να είναι μία από τις παρακάτω:

- **Pending** (σε αναμονή): είναι το πρώτο στάδιο και αναφέρεται στο διάστημα από την καταχώρηση της δράσης, δηλαδή της δημιουργίας της μέχρι και το πέρασμα στην κατάσταση «In Progress» ή «Cancelled». Πρόκειται για το διάστημα κατά το οποίο η δράση είναι διαθέσιμη στο κοινό για να συγκεντρώσει τις απαραίτητες συμμετοχές.
- **In Progress** (σε εξέλιξη): είναι το επόμενο στάδιο στο οποίο περνάει μία δράση αφού συγκεντρωθούν οι απαραίτητες συμμετοχές για την κάλυψη της επένδυσης και μέχρι την ολοκλήρωσή της.
- **Completed** (ολοκληρωμένη): πρόκειται για το τελικό στάδιο μιας δράσης που διαδέχεται το στάδιο «In Progress», αφού παρέλθει το χρονικό διάστημα της διάρκειας της δράσης. Αυτό σημαίνει ότι έχουν ολοκληρωθεί οι πληρωμές των τόκων και η επιστροφή του αρχικού κεφαλαίου από την επιχείρηση στους επενδυτές.
- **Canceled** (ακυρωμένη): είναι το στάδιο στο οποίο μεταβαίνει μία δράση όταν δεν καταφέρει να συγκεντρωθεί το απαιτούμενο κεφάλαιο μέχρι και την ημερομηνία έναρξής της.

Η μετάβαση από το αρχικό στάδιο στο στάδιο «In Progress» εκτελείται παράλληλα με την εισαγωγή των δεδομένων στον πίνακα payments. Για τη μετάβαση στα επόμενα στάδια, δεδομένου ότι στο σχεδιασμό της εφαρμογής δεν προβλέπεται η ύπαρξη διαχειριστή (administrator), θα πρέπει να πραγματοποιείται αυτόματα χωρίς τη μεσολάβηση κάποιας άλλης ενέργειας. Για το λόγο αυτό αναπτύχθηκαν οι Procedures **completed\_actions** και **canceled\_actions**, κάθε μία εκ των οποίων αναλαμβάνει τη μετάβαση στο στάδιο «Completed» ή «Canceled» αντίστοιχα. Για την εκτέλεση αυτών των Procedures προβλέφθηκε η δημιουργία κατάλληλων Events τα οποία θα εκτελούνται στον Scheduler κάθε μέρα στις 00:01, σύμφωνα με τις παρακάτω εντολές του Script 4.

```
01. SET GLOBAL event_scheduler = ON;
02.
03. CREATE EVENT check_completed
04. ON SCHEDULE
05. EVERY 1 DAY
06. STARTS (TIMESTAMP(CURRENT_DATE) + INTERVAL 1 DAY + INTERVAL 1 MINUTE)
07. DO
08.     CALL completed_actions();
09.
10. CREATE EVENT check_canceled
11. ON SCHEDULE
12. EVERY 1 DAY
13. STARTS (TIMESTAMP(CURRENT_DATE) + INTERVAL 1 DAY + INTERVAL 1 MINUTE)
14. DO
15.     CALL canceled_actions();
```

#### Script 4: Δημιουργία events στον scheduler

Η Procedure completed\_actions, η δημιουργία της οποίας βασίζεται στο Script 5, διατρέχει τις δράσεις που βρίσκονται στο στάδιο «In Progress» και θέτει την κατάστασή τους σε «Completed» εάν έχει παρέλθει η διάρκειά τους, καθώς εξυπακούεται ότι έχουν ολοκληρωθεί και οι πληρωμές από την επιχείρηση προς τους επενδυτές.

```
01. CREATE DEFINER='root'@'localhost' PROCEDURE `completed_actions`()
02. BEGIN
03.     DECLARE ids, dur INT;
04.     DECLARE done INT default 0;
05.     DECLARE sdate date;
06.     DECLARE cur CURSOR FOR SELECT Id,Duration,StartDate FROM action WHERE StatusId=2;
07.     DECLARE CONTINUE HANDLER FOR NOT FOUND SET DONE=1;
08.
09. OPEN cur;
10. segm: LOOP
11.     FETCH cur INTO ids, dur, sdate;
12.     IF done=1 THEN
13.         LEAVE segm;
14.     END IF;
15.     IF (adddate(sdate,INTERVAL dur YEAR)<=curdate()) THEN
16.         UPDATE action SET StatusId=3 WHERE Id=ids;
17.     END IF;
18. END LOOP segm;
19. CLOSE cur;
20. commit;
21. END
```

#### Script 5: Δημιουργία Procedure completed\_actions

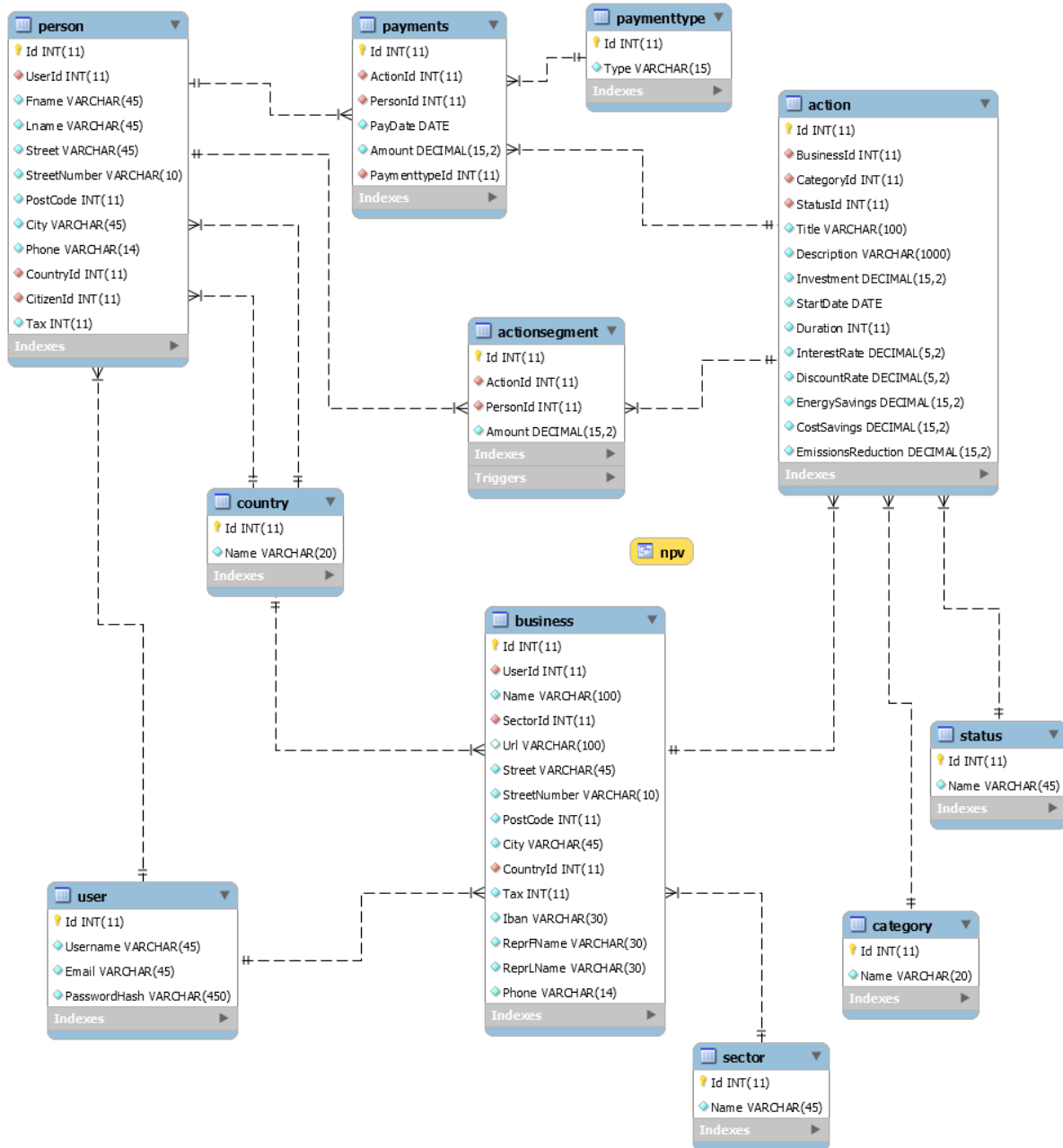
Από την πλευρά της, η Procedure canceled\_actions, όπως δημιουργήθηκε με το Script 6, διατρέχει τις δράσεις που βρίσκονται στο πρώτο στάδιο και θέτει την κατάστασή τους σε «Canceled» εάν έχει παρέλθει η ημερομηνία έναρξής τους χωρίς να έχει συγκεντρωθεί από τους επενδυτές το ύψος της επένδυσης που απαιτείται.

```
01. CREATE DEFINER=`root`@`localhost` PROCEDURE `canceled_actions`()
02. BEGIN
03. DECLARE ids INT;
04. DECLARE done INT default 0;
05. DECLARE sdate date;
06. DECLARE cur CURSOR FOR SELECT Id,StartDate FROM action WHERE StatusId=1;
07. DECLARE CONTINUE HANDLER FOR NOT FOUND SET DONE=1;
08.
09. OPEN cur;
10. segm: LOOP
11. FETCH cur INTO ids, sdate;
12. IF done=1 THEN
13. LEAVE segm;
14. END IF;
15. IF (sdate<curdate()) THEN
16. UPDATE action SET StatusId=4 WHERE Id=ids;
17. END IF;
18. END LOOP segm;
19. CLOSE cur;
20. commit;
21. END
```

Script 6: Δημιουργία Procedure canceled\_actions

### 3.1.5. Διάγραμμα ER της βάσης δεδομένων

Παρατίθεται το παρακάτω Διάγραμμα Οντοτήτων – Συσχετίσεων της βάσης δεδομένων που αναπτύχθηκε. Σε αυτό διακρίνονται η διάρθρωση των πινάκων και οι μεταξύ τους σχέσεις, τα PKs και FKs αλλά και σε γενικές γραμμές η λογική δομή της βάσης στην οποία στηρίζεται η λειτουργία της πλατφόρμας.



Διάγραμμα 7: ER Diagram

### 3.2. Δόμηση της εφαρμογής

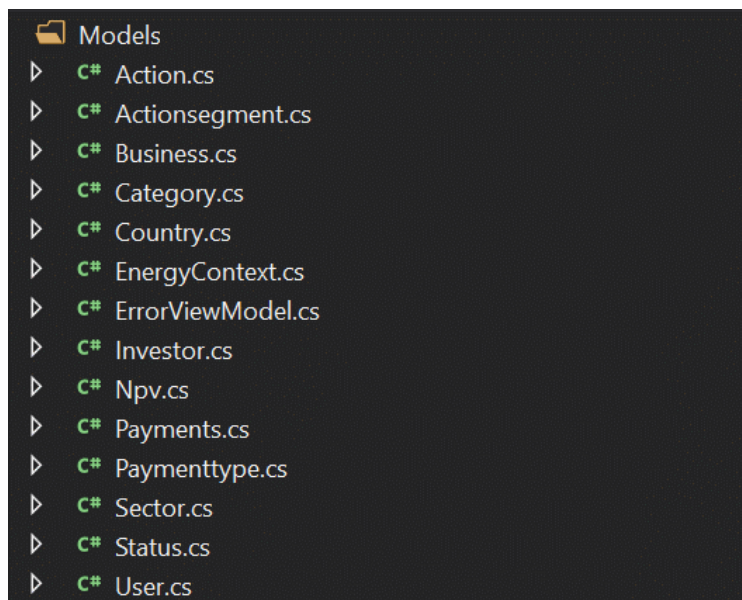
Για την ανάπτυξη της πλατφόρμας υιοθετήθηκε η προσέγγιση «Database First», σύμφωνα με την οποία, όπως έχει προαναφερθεί η κατασκευή της βάσης δεδομένων προηγείται της εφαρμογής. Στα πλαίσια αυτά, κατόπιν ολοκλήρωσης της κατασκευής της

βάσης δεδομένων και με τη χρήση του EF υλοποιήθηκε αυτή η προσέγγιση με την εκτέλεση των εντολών του Script 7.

```
>> dotnet add package Pomelo.EntityFrameworkCore.MySql
>> dotnet add package Microsoft.EntityFrameworkCore.Design
>> dotnet ef dbcontext scaffold "Server=localhost;Database=energy;User=root"
    Pomelo.EntityFrameworkCore.MySql -o Model -c "EnergyContext"
```

#### Script 7: Χρήση του Entity Framework

Χρησιμοποιώντας το Developer PowerShell Terminal, που μας προσφέρει το VS, με τις δύο πρώτες εντολές εισάγονται τα Nuget Packages, Pomelo Entity Framework Core MySql και Microsoft Entity Framework Core Design, τα οποία απαιτούνται για τη δημιουργία των αρχικών μοντέλων στα πλαίσια του MVC και τη σύνδεση της εφαρμογής με τη βάση. Με την τρίτη εντολή εκτελείται scaffold στην βάση που έχει δημιουργηθεί και μέσω του πρώτου Nuget κατασκευάζεται το μοντέλο EnergyContext που αντιπροσωπεύει την βάση, ενώ παράλληλα δημιουργούνται και τα αντίστοιχα μοντέλα για κάθε πίνακα και view που υπάρχει σε αυτή, Εικόνα 7.



Εικόνα 7: Δημιουργία Μοντέλων μέσω του EF



### 3.2.1. Εγγραφή και είσοδος χρηστών

Ένα από τα βασικότερα συστατικά στοιχεία μιας εφαρμογής είναι η διαδικασία εγγραφής και εισόδου των χρηστών, καθώς και οι διάφοροι ρόλοι που μπορεί να έχουν σε αυτή. Δεδομένου ότι βάσει σχεδιασμού της πλατφόρμας δεν προβλέπεται η δημιουργία χρήστη με δικαιώματα διαχειριστή, καθώς για τους σκοπούς ανάπτυξής της δεν κρίνεται απαραίτητο, η αναφορά πλέον σε χρήστες θα απευθύνεται είτε σε επιχειρήσεις είτε σε επενδυτές.

Για την υλοποίηση της διαδικασίας εγγραφής και εισόδου, γίνεται χρήση του ASP.NET Core Identity, όπως έχει ήδη προαναφερθεί. Σε πρώτη φάση δημιουργείται το `Extention UserStore` που είναι απαραίτητο για την αποθήκευση και τη διαχείριση των λογαριασμών των χρηστών και εφαρμόζει το `Interface IUserStore`. Σε αυτό το σημείο αξίζει να σημειωθεί ότι η απονομή ρόλων στους χρήστες δεν υλοποιείται στα πλαίσια της εφαρμογής και για το λόγο αυτό η κατασκευή του αντίστοιχου `Extention` για το `Interface IRoleStore` δεν πραγματοποιήθηκε. Από την άλλη, στην κλάση `UserStore` υλοποιούνται οι απαραίτητες μέθοδοι για την δημιουργία, ενημέρωση και ανάκτηση των χρηστών στη βάση δεδομένων, εκείνες δηλαδή που είναι αναγκαίες για τη λειτουργικότητα της εφαρμογής. Οι μέθοδοι της κλάσης χρησιμοποιούνται από τους αντίστοιχους `managers` του Identity (`userManager` και `signInManager`) στον `Controller` που έχει δημιουργηθεί για τον χειρισμό των `Views` που θα αναλάβουν την είσοδο και εγγραφή των χρηστών, όπως αναλύεται σε επόμενη ενότητα του κεφαλαίου.

Για τη διαμόρφωση του Identity, όπως οι απαιτήσεις του password και του username των χρηστών, προστέθηκαν τα αντίστοιχα `services` στη μέθοδο `ConfigureServices` της κλάσης `Startup` της εφαρμογής, όπως στο παρακάτω `Script`. Αντίστοιχα, παραμετροποιήθηκε η μέθοδος `Configure` της ίδιας κλάσης. Η μέθοδος `ConfigureServices` περιλαμβάνει τα `services` που θα χρησιμοποιεί η εφαρμογή, ενώ η μέθοδος `Configure` χρησιμοποιείται για τη δημιουργία μιας οδού επεξεργασίας των αιτημάτων της. Και οι δύο μέθοδοι καλούνται από το ASP.NET Core runtime κατά την εκκίνηση της εφαρμογής.

```
01. public void ConfigureServices(IServiceCollection services)
02. {
03.     services.AddDbContext<EnergyContext>();
04.     services.AddControllersWithViews();
05.     services.AddIdentity<User, UserRole>().AddDefaultTokenProviders();
06.     services.AddTransient<IUserStore<User>, UserStore>();
07.     services.AddTransient<IRoleStore<UserRole>, RoleStore>();
08.     services.Configure<IdentityOptions>(options =>
09.     {
10.         // Password settings.
11.         options.Password.RequireDigit = true;
12.         options.Password.RequireLowercase = false;
13.         options.Password.RequireNonAlphanumeric = false;
14.         options.Password.RequireUppercase = true;
15.         options.Password.RequiredLength = 6;
16.         options.Password.RequiredUniqueChars = 1;
17.
18.         // User settings.
19.         options.User.AllowedUserNameCharacters =
20.         "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789-._@+";
21.         options.User.RequireUniqueEmail = false;
22.     });
23.     services.ConfigureApplicationCookie(options =>
24.     {
25.         options.Cookie.HttpOnly = true;
26.         options.LoginPath = "/Account/Login";
27.     });
28. }
29.
30. // This method gets called by the runtime. Use this method to configure the HTTP request pipeline.
31. public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
32. {
33.     if (env.IsDevelopment())
34.     {
35.         app.UseDeveloperExceptionPage();
36.     }
37.     else
38.     {
39.         app.UseExceptionHandler("/Home/Error");
40.         app.UseHsts();
41.     }
42.     app.UseHttpsRedirection();
43.     app.UseStaticFiles();
44.     app.UseRouting();
45.     app.UseAuthentication();
46.     app.UseAuthorization();
47.     app.UseEndpoints(endpoints =>
48.     {
49.         endpoints.MapControllerRoute(
50.             name: "default",
51.             pattern: "{controller=Home}/{action=Index}/{id?}");
52.     });
53. }
```

Script 8: Διαμόρφωση της κλάσης Startup

### 3.2.2. Υλοποίηση του προτύπου MVC

Η πλατφόρμα έχει κατασκευαστεί σύμφωνα με το πρότυπο MVC. Ο τρόπος λειτουργίας μιας εφαρμογής βάσει του προτύπου αυτού περιλαμβάνει τα εξής στάδια:

- Αρχικά ο χρήστης βλέπει το UI της εφαρμογής μέσω ενός View, στο οποίο βρίσκονται τα διάφορα στοιχεία αλληλεπίδρασης, όπως σύνδεσμοι, μενού, φόρμες, κουμπιά κ.ά.

- Στη συνέχεια μέσω κάποια αλληλεπίδρασης του χρήστη με το View, πατώντας για παράδειγμα κάποιον σύνδεσμο, η πληροφορία αυτή της αλληλεπίδρασης μεταφέρεται στον αντίστοιχο Controller που είναι συσχετισμένος με το αντίστοιχο View.
- Από την πλευρά του ο Controller, επεξεργάζεται τις πληροφορίες του χρήστη καθώς και των αντίστοιχων Models που είναι συνυφασμένα με την ενέργεια αυτή του χρήστη. Αυτά τα Models, μπορεί να αφορούν κλάσεις της βάσης δεδομένων ή άλλες κλάσεις που δημιουργήθηκαν και απαιτούνται για τη λειτουργία της εφαρμογής.
- Σε τελικό στάδιο, αφού ενημερωθούν τα Models και εκτελεστούν οι εντολές στον Controller, ο τελευταίος ενημερώνει το νέο View που θα παρουσιάσει στον χρήστη και αποστέλλει τις αναγκαίες πληροφορίες.

#### 3.2.2.1. Routing

Στο πρότυπο MVC η πλοήγηση μεταξύ των σελίδων γίνεται με τη χρήση URL, το οποίο αντιστοιχίζεται σε ένα συγκεκριμένο Action ενός Controller και περιλαμβάνει την απαραίτητη πληροφορία για την επιλογή του Action, του Controller αλλά και των παραμέτρων που θα χρησιμοποιηθούν κατά την κλήση του Action. Η συσχέτιση μεταξύ των URLs και των Controllers ορίζεται ως Routing. Στην εφαρμογή μας η παραμετροποίηση του Routing με τη χρήση Endpoints και του μοτίβου που θα αναγνωρίζει περιλαμβάνεται στη μέθοδο Configure της κλάσης Startup, όπως στο Script 8. Βάσει αυτού ορίζεται ότι μετά το domain της εφαρμογής θα ακολουθεί το όνομα του Controller και έπειτα το όνομα του Action του Controller που καλείται, ενώ μετά το τελευταίο υπάρχει η δυνατότητα συμπλήρωσης των όποιων παραμέτρων απαιτούνται για την εκτέλεσή του.

#### 3.2.2.2. Controllers

Όπως έχει ήδη αποτυπωθεί, οι Controllers αποτελούν το διαχειριστή του UI της εφαρμογής, αλλά και το συνδεδετικό κρίκο ανάμεσα στα Views και τα Models. Μέσω, λοιπόν,

του Routing καλείται ένας Controller και έπειτα εκτελείται ένα Interface IActionResult με βάση το input του χρήστη, το οποίο μπορεί να επιστρέφει ένα View, ή να καλεί ένα άλλο IActionResult στον ίδιο ή σε άλλον Controller.

Για την πληρέστερη και πιο δομημένη υλοποίηση της εφαρμογής μας, δημιουργήθηκαν δύο διαφορετικοί Controllers. Ο AccountController θα διαχειρίζεται τα Views που αφορούν την εγγραφή, την είσοδο και την έξοδο των χρηστών από την εφαρμογή, ενώ ο HomeController τα λοιπά Views που αφορούν την κύρια λειτουργία της εφαρμογής. Και οι δύο Controllers περιλαμβάνουν από έναν constructor αντίστοιχα, ο οποίος χρησιμεύει για την αρχικοποίηση των μεταβλητών που αναφέρονται στην βάση δεδομένων και τους αναγκαίους managers που χρησιμοποιούνται από τις μεθόδους αυτών.

Τα Actions που δημιουργήθηκαν στον AccountController είναι τα παρακάτω:

- *IActionResult Register*: Διευκολύνει τον χρήστη να επιλέξει το είδος της εγγραφής που επιθυμεί.
- *IActionResult RegisterBusiness*: Παρουσιάζει στο χρήστη μία φόρμα για τη συμπλήρωση των απαραίτητων στοιχείων προκειμένου να ολοκληρώσει την εγγραφή της επιχείρησής του στην πλατφόρμα.
- *IActionResult RegisterInvestor*: Παρουσιάζει στο χρήστη μία φόρμα για τη συμπλήρωση των απαραίτητων στοιχείων προκειμένου να ολοκληρώσει την εγγραφή του ως επενδυτή στην πλατφόρμα.
- *async Task<IActionResult> RegisterBusiness(BusinessUserRegister model)*: Είναι ένα ασύγχρονο Task που δέχεται ως παράμετρο ένα Model και επιστρέφει ένα IActionResult κατόπιν υποβολής της φόρμας εγγραφής επιχείρησης από το χρήστη.
- *async Task<IActionResult> RegisterInvestor(InvestorUserRegister model)*: Είναι ένα ασύγχρονο Task που δέχεται ως παράμετρο ένα Model και επιστρέφει ένα IActionResult κατόπιν υποβολής της φόρμας εγγραφής επενδυτή από το χρήστη.
- *IActionResult Login*: Είναι το Action που παρουσιάζει στο χρήστη μία φόρμα για την εισαγωγή των διαπιστευτηρίων του προκειμένου να εισέλθει στην πλατφόρμα.

- *async Task<IActionResult> Login(LoginProfile model)*: Είναι ένα ασύγχρονο Task που δέχεται ως παράμετρο ένα Model και επιστρέφει ένα IActionResult, αναλαμβάνοντας την είσοδο του χρήστη στην πλατφόρμα.
- *async Task<IActionResult> Logout*: Είναι ένα ασύγχρονο Task που αναλαμβάνει την έξοδο του συνδεδεμένου χρήστη από την πλατφόρμα.

Αντίστοιχα, τα Actions που δημιουργήθηκαν στον HomeController έχουν ως εξής:

- *IActionResult Index*: Παρουσιάζει την αρχική σελίδα της πλατφόρμας.
- *IActionResult Actions*: Παρουσιάζει την σελίδα με όλες τις καταχωρημένες δράσεις εξοικονόμησης ενέργειας.
- *IActionResult Action(int actionid)*: Παρουσιάζει τη σελίδα όπου αναλύεται μία δράση ως προς τα χαρακτηριστικά της.
- *IActionResult BusinessProfile*: Είναι η σελίδα που περιλαμβάνει το προφίλ μιας συνδεδεμένης επιχείρησης.
- *IActionResult InvestorProfile*: Είναι η σελίδα που περιλαμβάνει το προφίλ ενός συνδεδεμένου επενδυτή.
- *IActionResult RegisterAction*: Παρουσιάζει στο χρήστη (επιχείρηση) μία φόρμα για τη συμπλήρωση των απαραίτητων στοιχείων προκειμένου να καταχωρήσει μία νέα δράση εξοικονόμησης ενέργειας.
- *async Task<bool> RegisterAction (ActionRegister model)*: Είναι ένα ασύγχρονο Task που δέχεται ως παράμετρο ένα Model και επιστρέφει true ή false, επιχειρώντας να καταχωρήσει στη βάση δεδομένων μία νέα δράση.
- *async Task<bool> RegisterActionSegment (int investorid, int actionid, decimal amount)*: Είναι ένα ασύγχρονο Task που δέχεται ως παράμετρο ένα Model και επιστρέφει true ή false, αναλαμβάνοντας την καταχώρηση μιας συμμετοχής ενός επενδυτή σε μία δράση.

### 3.2.2.3. Models

Πέραν των ήδη υπαρχόντων μοντέλων, όπως προέκυψαν μέσω του EF και αντανακλούν τους πίνακες και το View της βάσης δεδομένων, κατασκευάστηκαν νέα μοντέλα έτσι ώστε να γίνεται ευκολότερη και πιο δομημένη η μεταφορά πληροφορίας μεταξύ των Controllers και των Views της εφαρμογής.

Ειδικότερα, δημιουργήθηκαν τα εξής μοντέλα:

- **ActionRegister** : Χρησιμεύει στη μεταφορά πληροφορίας κατά την καταχώρηση μιας δράσης (Script 9).

```
01. namespace EnergyEfficiencyActions.Models
02. {
03.     public partial class ActionRegister
04.     {
05.         [Required]
06.         public Action action { get; set; }
07.
08.         public Dictionary<int, string> Categories { get; set; }
09.     }
10. }
```

Script 9: Model ActionRegister

- **ActionSearch** : Χρησιμεύει στη μεταφορά πληροφορίας κατά την αναζήτηση των δράσεων (Script 10).

```
01. namespace EnergyEfficiencyActions.Models
02. {
03.     public partial class ActionSearch
04.     {
05.         [Required]
06.         public IEnumerable<Action> actions { get; set; }
07.
08.         public Dictionary<int, string> ActionCategories { get; set; }
09.     }
10. }
```

Script 10: Model ActionSearch

- **ActionView** : Χρησιμεύει στη μεταφορά πληροφορίας κατά την προβολή της σελίδας μιας δράσης (Script 11).

```
01. namespace EnergyEfficiencyActions.Models
02. {
03.     public partial class ActionView
04.     {
05.         [Required]
06.         public Action action { get; set; }
07.         public int? investorId { get; set; }
08.         public int? businessId { get; set; }
09.         public decimal restAmount { get; set; }
10.         public IEnumerable<Npv> npvViews { get; set; }
11.     }
12. }
```

Script 11: Model ActionView

- **BusinessProfile** : Χρησιμοποιείται για τη μεταφορά πληροφορίας κατά την προβολή του προφίλ μιας επιχείρησης (Script 12).

```
01. namespace EnergyEfficiencyActions.Models
02. {
03.     public partial class BusinessProfile
04.     {
05.         [Required]
06.         public Business business { get; set; }
07.         public IEnumerable<Action> actions { get; set; }
08.         public IEnumerable<Payments> payments { get; set; }
09.         public CountriesSectorsList countriesSectorsList { get; set; }
10.         public Dictionary<int, string> actionCategories { get; set; }
11.     }
12. }
```

Script 12: Model BusinessProfile

- **BusinessUserRegister**: Χρησιμοποιείται για τη μεταφορά πληροφορίας κατά την εγγραφή μιας επιχείρησης (Script 13).

```
01. namespace EnergyEfficiencyActions.Models
02. {
03.     public partial class BusinessUserRegister
04.     {
05.         [Required]
06.         public Business Business { get; set; }
07.
08.         public CountriesSectorsList countriesSectorsList { get; set; }
09.
10.         [Required]
11.         [MaxLength(45, ErrorMessage = "Username maximum 45 Characters")]
12.         public string Username { get; set; }
13.
14.         [Required(ErrorMessage = "Email is required")]
15.         [DataType(DataType.EmailAddress, ErrorMessage = "Email is invalid")]
16.         [MaxLength(45, ErrorMessage = "Email maximum 45 Characters")]
17.         public string Email { get; set; }
18.
19.         [Required]
20.         public string Password { get; set; }
21.
22.         [Required]
23.         [Compare("Password",
24.             ErrorMessage="Password and Confirmation Password do not match")]
25.         public string ConfirmPassword { get; set; }
26.     }
27. }
```

Script 13: Model BusinessUserRegister

- **InvestorProfile:** Χρησιμοποιείται για τη μεταφορά πληροφορίας κατά την προβολή του προφίλ ενός επενδυτή (Script 14).

```
01. namespace EnergyEfficiencyActions.Models
02. {
03.     public partial class InvestorProfile
04.     {
05.         [Required]
06.         public Investor investor { get; set; }
07.         public IEnumerable<Action> actions { get; set; }
08.         public IEnumerable<Actionsegment> actionsegments { get; set; }
09.         public IEnumerable<Payments> payments { get; set; }
10.         public CountriesSectorsList CountriesSectorsList { get; set; }
11.         public Dictionary<int, string> ActionCategories { get; set; }
12.     }
13. }
```

Script 14: Model InvestorProfile

- **InvestorUserRegister:** Χρησιμοποιείται για τη μεταφορά πληροφορίας κατά την εγγραφή ενός επενδυτή (Script 15).

```
01. namespace EnergyEfficiencyActions.Models
02. {
03.     public partial class InvestorUserRegister
04.     {
05.         [Required]
06.         public Investor Investor { get; set; }
07.
08.         public CountriesSectorsList CountriesSectorsList { get; set; }
09.
10.         [Required]
11.         [MinLength(4, ErrorMessage = "Username minimum 4 Characters")]
12.         [MaxLength(45, ErrorMessage = "Username maximum 45 Characters")]
13.         public string Username { get; set; }
14.
15.         [Required(ErrorMessage = "Email is required")]
16.         [DataType(DataType.EmailAddress, ErrorMessage = "Email is invalid")]
17.         [MaxLength(45, ErrorMessage = "Email maximum 45 Characters")]
18.         public string Email { get; set; }
19.
20.         [Required]
21.         public string Password { get; set; }
22.
23.         [Required]
24.         [Compare("Password",
25.             ErrorMessage="Password and Confirmation Password do not match")]
26.         public string ConfirmPassword { get; set; }
27.     }
28. }
```

Script 15: Model InvestorUserRegister

- **CountriesSectorsList:** Χρησιμεύει για την αποθήκευση σε λίστες των διαφόρων χωρών και τομέων δραστηριοποίησης των επιχειρήσεων (Script 16).



```
01. namespace EnergyEfficiencyActions.Models
02. {
03.     public partial class CountriesSectorsList
04.     {
05.         public Dictionary<int,string> Countries { get; set; }
06.         public Dictionary<int, string> Sectors { get; set; }
07.     }
08. }
```

Script 16: Model CountriesSectorsList

- **LoginProfile:** Χρησιμεύει στη μεταφορά πληροφορίας κατά την είσοδο ενός χρήστη στην εφαρμογή (Script 17).

```
01. namespace EnergyEfficiencyActions.Models
02. {
03.     public partial class LoginProfile
04.     {
05.         [Required]
06.         public string Username { get; set; }
07.
08.         [Required]
09.         public string Password { get; set; }
10.
11.         public string returnUrl { get; set; }
12.     }
13. }
```

Script 17: Login Profile

#### 3.2.2.4. Views

Τα Views στο ASP.NET Core είναι αρχεία .cshtml που χρησιμοποιούν τη γλώσσα C# με σήμανση Razor. Η σύνταξη του Razor παρέχει έναν απλό και εύχρηστο τρόπο για την απόδοση HTML περιεχομένου βάσει του View, συνδυάζοντας την χρήση της C# με HTML και jQuery, παράγοντας ιστοσελίδες πλήρως συμβατές με την έκδοση HTML5.

Τα αντικείμενα που είναι απαραίτητα σε ένα View, όπως για παράδειγμα ένα Model, επεξεργάζονται από τον Controller και τίθενται ως ορίσματα κατά την κλήση του View μέσω ενός Action του Controller. Συνεπώς, κατά τη χρήση ενός Model ως ορίσματος, θα πρέπει το ίδιο να ορίζεται μέσα στο View ώστε να μπορεί να χρησιμοποιηθεί για την απεικόνιση της πληροφορίας.

Οι περισσότερες εφαρμογές, όπως και η παρούσα, χρησιμοποιούν ένα κοινό σχέδιο (Layout), επιφορτισμένο να παρέχει στο χρήστη μία συνεπή εμπειρία περιεχομένου κατά την πλοήγηση στην εφαρμογή μεταξύ των σελίδων. Το σχέδιο αυτό στην περίπτωση της

πλατφόρμας περιλαμβάνει κοινά στοιχεία εμφάνισης των σελίδων της, όπως το header, το μενού πλοήγησης και το footer. Μέσα στον κώδικα που περιγράφει το Layout και με την εντολή `@RenderBody()` καλείται το αντίστοιχο View που πρέπει να φορτωθεί, ενώ με την εντολή `@RenderSection()` δίνεται η δυνατότητα δημιουργίας τομέων εντός του Layout οι οποίοι μπορούν να ορίζονται εντός του View.

Είθισται, όπως τα Views ομαδοποιούνται σε διαφορετικούς φακέλους με την ονομασία του Controller στον οποίο αντιστοιχούν. Έτσι ακριβώς και στην εφαρμογή, υπάρχουν δύο φάκελοι Account και Home, καθένας εκ των οποίων περιλαμβάνει τα Views του αντίστοιχου Controller.

Ειδικότερα, τα Views, που δημιουργήθηκαν στον φάκελο Account είναι:

- **Login:** Η σελίδα σύνδεσης ενός χρήστη στην πλατφόρμα.
- **Register:** Βοηθητική σελίδα για την επιλογή καταχώρησης επιχείρησης ή επενδυτή ως νέου χρήστη στην πλατφόρμα.
- **RegisterBusiness:** Η σελίδα καταχώρησης των στοιχείων για την εγγραφή μιας νέας επιχείρησης.
- **RegisterInvestor:** Η σελίδα καταχώρησης των στοιχείων για την εγγραφή ενός νέου επενδυτή.

Αντίστοιχα, στον φάκελο Home δημιουργήθηκαν τα παρακάτω Views:

- **Index:** Η αρχική σελίδα της πλατφόρμας.
- **Action:** Η σελίδα παρουσίασης των στοιχείων μιας δράσης.
- **Actions:** Η σελίδα όπου εμφανίζονται όλες οι δράσεις που έχουν καταχωρηθεί στην πλατφόρμα.
- **RegisterAction:** Η σελίδα καταχώρησης μιας νέας δράσης.
- **BusinessProfile:** Η σελίδα όπου εμφανίζεται το προφίλ της συνδεδεμένης επιχείρησης.
- **InvestorProfile:** Η σελίδα όπου εμφανίζεται το προφίλ του συνδεδεμένου επενδυτή.

## 4. ΠΑΡΟΥΣΙΑΣΗ ΤΟΥ UI ΚΑΙ ΤΗΣ ΛΕΙΤΟΥΡΓΙΑΣ ΤΗΣ ΠΛΑΤΦΟΡΜΑΣ

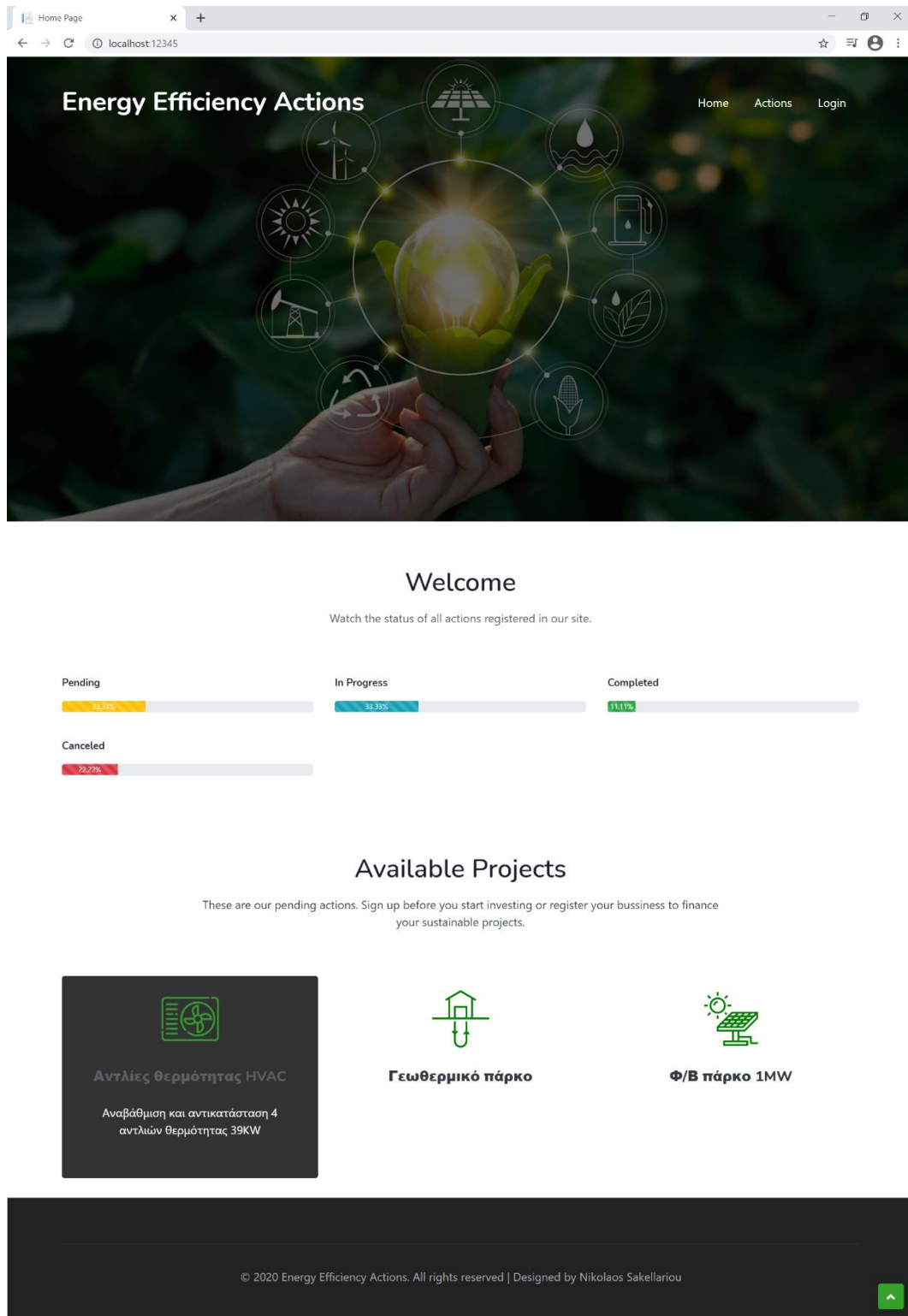
Στόχος του UI της εφαρμογής είναι να παρέχει μία απλή, εύκολη, γρήγορη και ευχάριστη εμπειρία στους χρήστες της. Για το λόγο αυτό έχει συνδυαστεί η χρήση της HTML για τη δημιουργία των ιστοσελίδων, με CSS, JavaScript, JQuery και Ajax παρέχοντας μια ολοκληρωμένη, χρηστική και εύληπτη παρουσίαση της λειτουργικότητας της πλατφόρμας.

Στο άνω μέρος της κάθε σελίδας βρίσκεται το μενού πλοήγησης, το οποίο είναι δυναμικό και μεταβάλλεται ανάλογα με το αν ο επισκέπτης είναι συνδεδεμένος ή όχι και στην περίπτωση του συνδεδεμένου χρήστη ανάλογα με το αν ο λογαριασμός του αντιστοιχεί σε επιχείρηση ή επενδυτή.

### 4.1. Αρχική σελίδα

Η αρχική σελίδα καλωσορίζει τους επισκέπτες της πλατφόρμας, παρέχοντας μια πρώτη επαφή με τις δράσεις εξοικονόμησης ενέργειας που έχουν καταχωρηθεί στην πλατφόρμα. Ειδικότερα, γίνεται αντιληπτή η κατάσταση όλων των καταχωρημένων δράσεων, ποιο ποσοστό δηλαδή εξ αυτών είναι σε αναμονή, σε εξέλιξη, έχουν ολοκληρωθεί ή ακυρωθεί. Ακόμη, δίνεται η δυνατότητα προβολής των δράσεων που είναι σε εξέλιξη και απαιτούνται ενέργειες από τους επενδυτές για τη χρηματοδότησή τους, ταξινομημένες ως προς την ημερομηνία έναρξης.

Οι επιλογές πλοήγησης από την αρχική σελίδα εντοπίζονται είτε μέσω του μενού πλοήγησης στο άνω μέρος της σελίδας, είτε μέσω των διαθέσιμων δράσεων στο τέλος της σελίδας, όπου επιλέγοντας μία οποιαδήποτε δράση γίνεται μεταφορά στη αντίστοιχη σελίδα της. Από το μενού υπάρχουν επιλογές για μεταφορά στην καρτέλα των δράσεων ή στη σελίδα όπου πραγματοποιείται η είσοδος στην πλατφόρμα. Η εμφάνιση της αρχικής σελίδας έχει όπως στην Εικόνα 8.

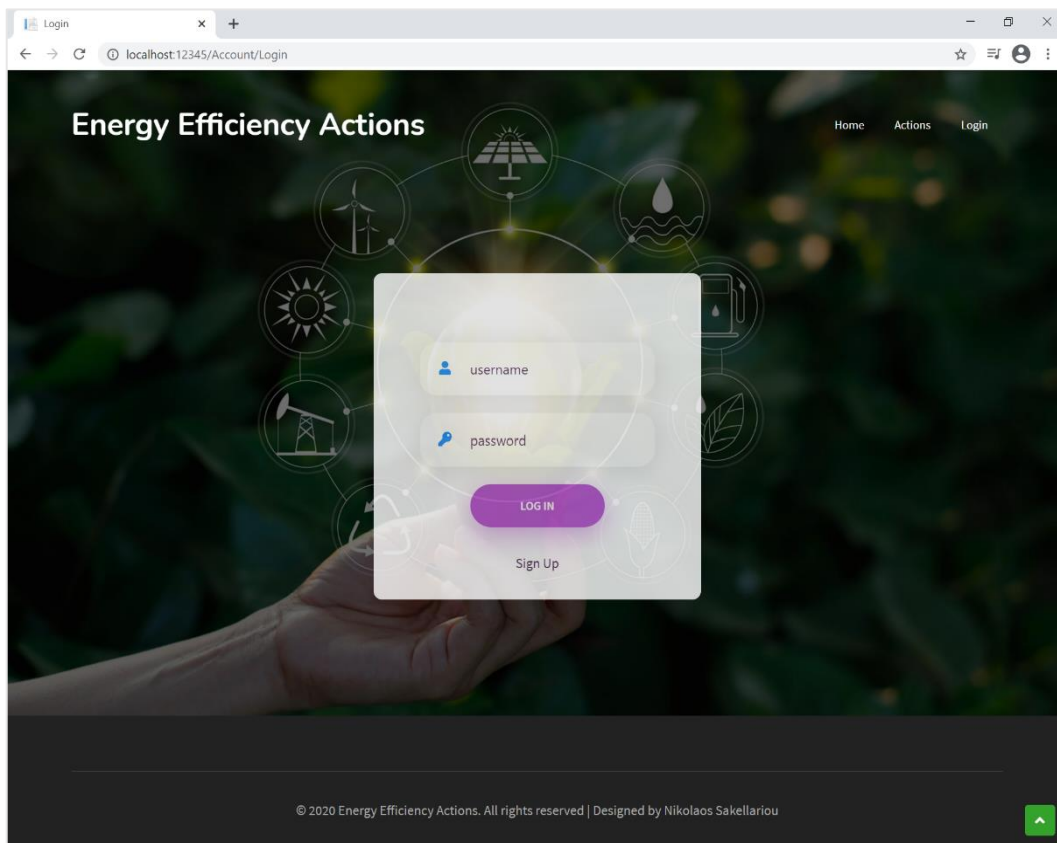


Εικόνα 8: Αρχική σελίδα

## 4.2. Σύνδεση Χρηστών

Επιλέγοντας από το μενού πλοήγησης την είσοδο στην πλατφόρμα (Login) γίνεται πλοήγηση στην αντίστοιχη σελίδα, όπως στην Εικόνα 9. Εδώ ο χρήστης μπορεί να εισάγει τα διαπιστευτήριά του και να συνδεθεί στην πλατφόρμα. Στην περίπτωση που δε διαθέτει λογαριασμό μπορεί να επιλέξει τον υπερσύνδεσμο «Sign Up» και να μεταφερθεί στη σελίδα εγγραφής νέου χρήστη.

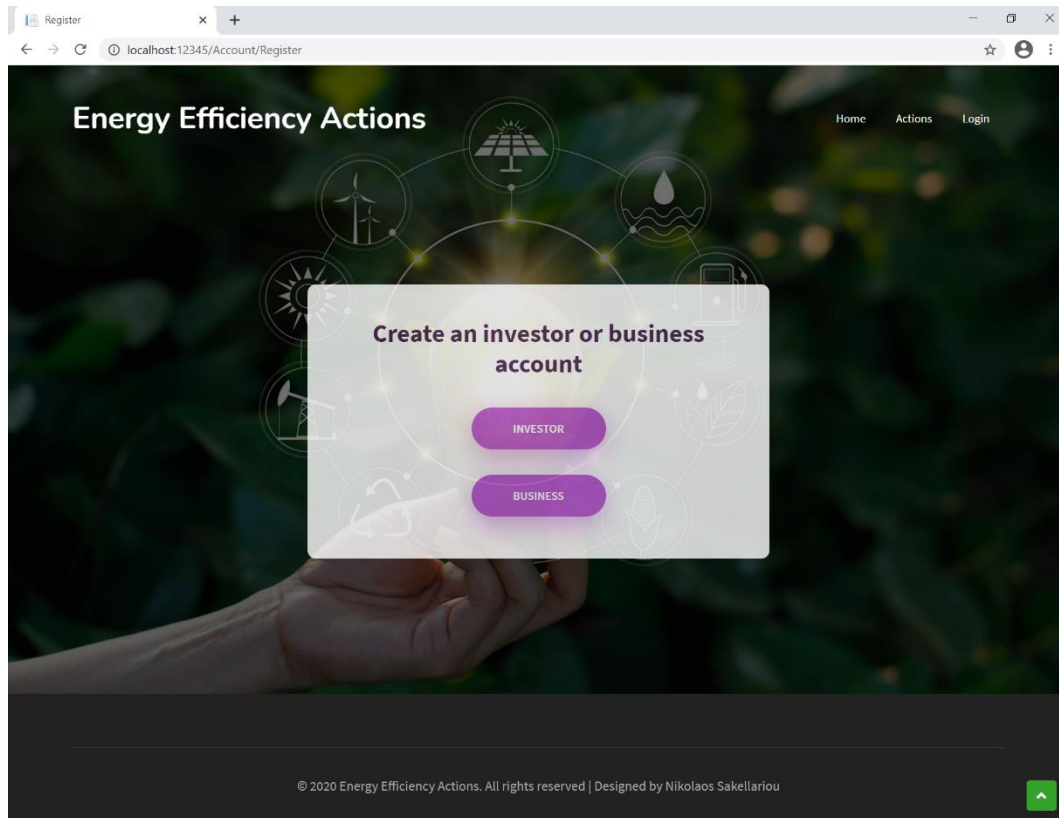
Εαν τα διαπιστευτήρια που εισάγει ο χρήστης δεν αντιστοιχούν σε λογαριασμό τότε ένα pop-up παράθυρο ενημερώνει το χρήστη για την αποτυχία εισόδου και την ανάγκη εισαγωγής έγκυρων διαπιστευτηρίων.



Εικόνα 9: Σελίδα σύνδεσης

### 4.3. Εγγραφή Χρηστών

Με τη χρήση του υπερσυνδέσμου «Sign Up» της σελίδας σύνδεσης ο επισκέπτης μεταφέρεται στην βοηθητική σελίδα εγγραφής χρηστών (Εικόνα 10) προκειμένου να επιλέξει αν επιθυμεί την δημιουργία λογαριασμού επιχείρησης ή επενδυτή.



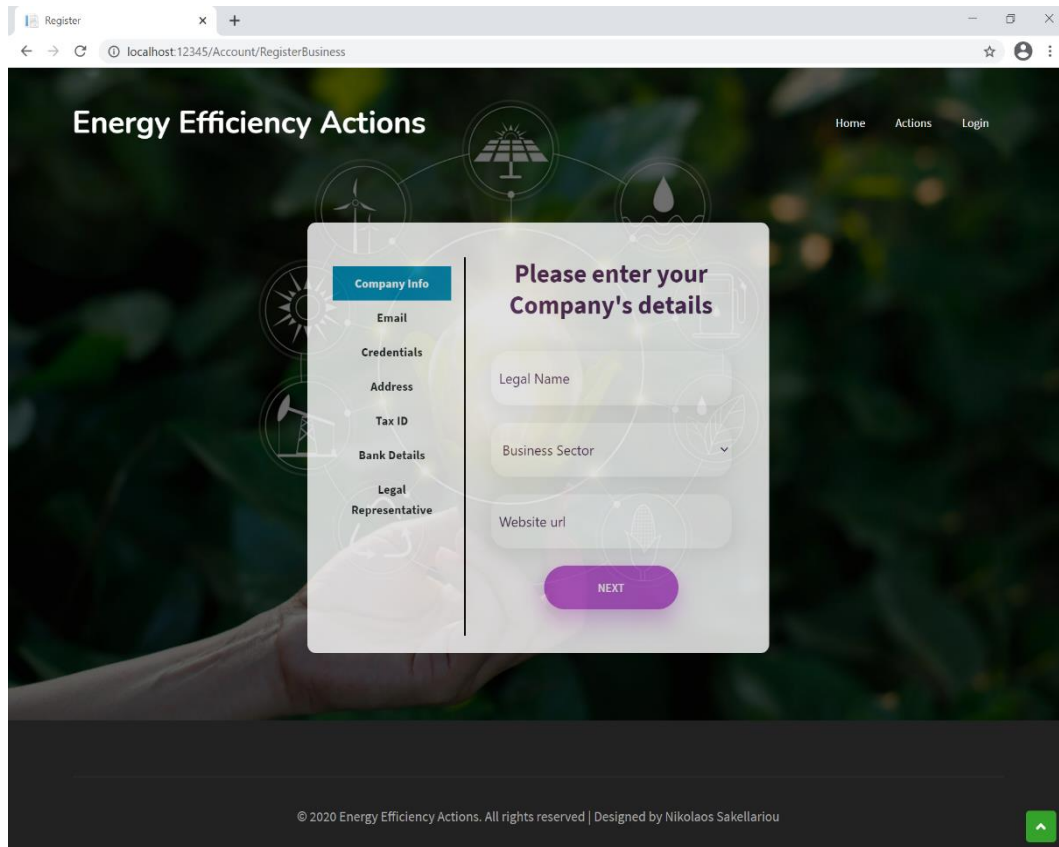
Εικόνα 10: Βοηθητική σελίδα εγγραφής χρηστών

#### 4.3.1. Καταχώρηση Επιχείρησης

Ακολουθώντας τον σύνδεσμο της επιχείρησης από την προηγούμενη βοηθητική σελίδα εγγραφής ο επισκέπτης μεταφέρεται στη σελίδα όπου θα του ζητηθούν τα απαραίτητα στοιχεία για την ολοκλήρωση της δημιουργίας λογαριασμού μιας επιχείρησης. Η φόρμα που καλείται να συμπληρώσει ο επισκέπτης είναι διαβαθμισμένη σε 7 καρτέλες (tabs), στις οποίες μπορεί να πλοηγηθεί είτε μέσω της αριστερής πλευράς της φόρμας, όπου εμφανίζονται οι καρτέλες

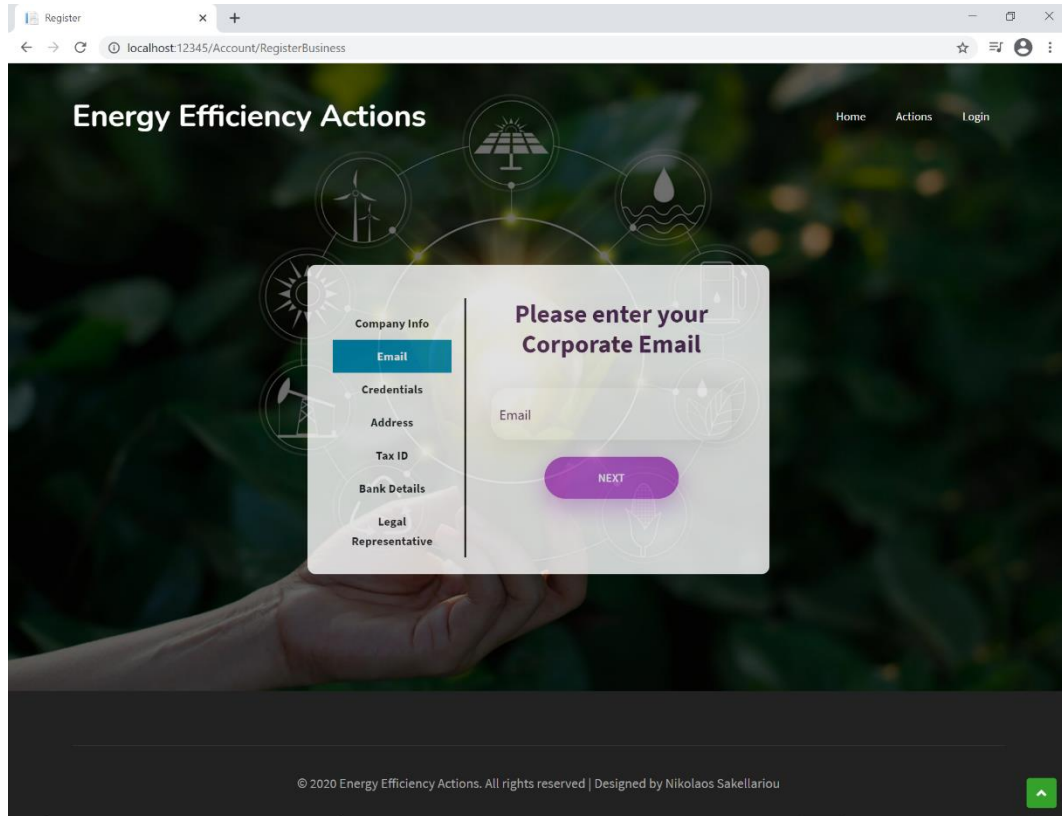
είτε με τα κουμπιά «Next», με τη χρήση των οποίων γίνεται μεταφορά στην αμέσως επόμενη καρτέλα.

Στην πρώτη καρτέλα, όπως φαίνεται στην Εικόνα 11, ο επισκέπτης συμπληρώνει το όνομα της νομικής οντότητας, τον τομέα στον οποίο δραστηριοποιείται (επιλέγοντας από ένα dropdown list) καθώς και το url της ιστοσελίδας της επιχείρησης (αν υπάρχει).

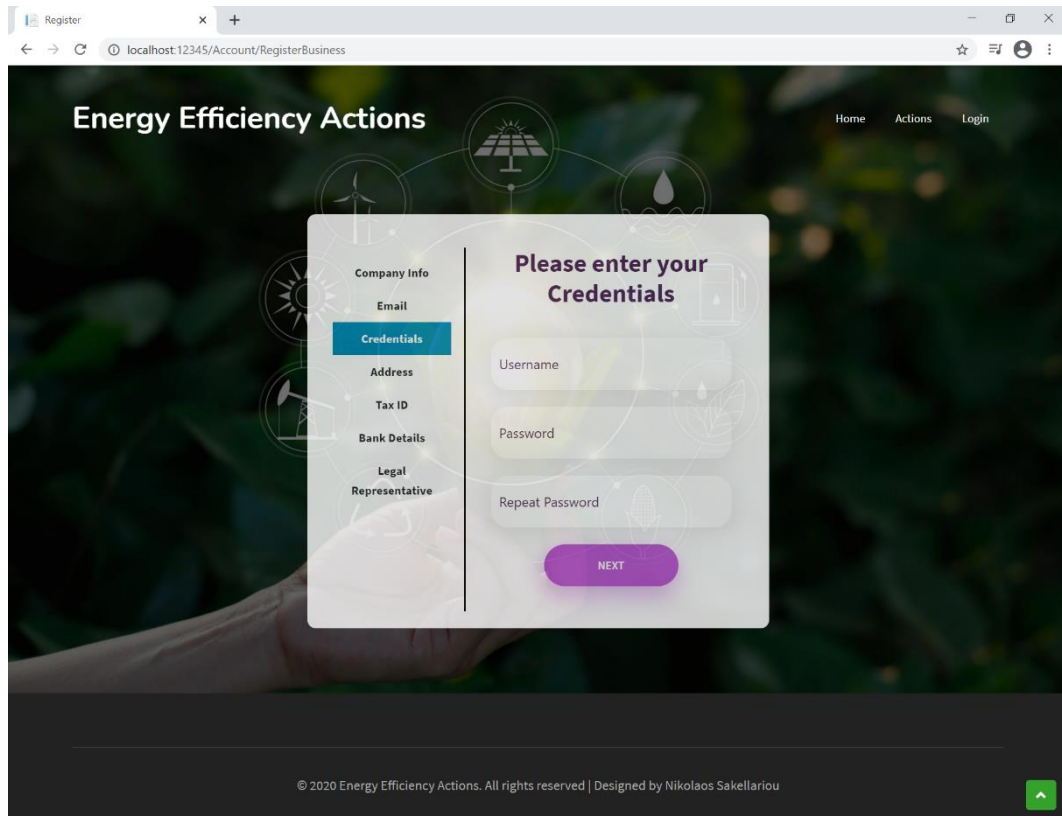


Εικόνα 11: Εγγραφή Επιχείρησης - Γενικά Στοιχεία

Στη δεύτερη καρτέλα, όπως στην Εικόνα 12, ο επισκέπτης καλείται να συμπληρώσει το εταιρικό email της επιχείρησης ενώ με την τρίτη καρτέλα, όπως εμφανίζεται στην Εικόνα 13, συμπληρώνονται τα στοιχεία σύνδεσης του χρήστη που θα δημιουργηθεί, δηλαδή το όνομα χρήστη και ο κωδικός πρόσβασης. Το όνομα χρήστη θα πρέπει να είναι μοναδικό, ενώ ο κωδικός πρόσβασης να πληροί τις ελάχιστες απαιτήσεις που τέθηκαν στη μέθοδο ConfigureServices της κλάσης Startup (Script 8).



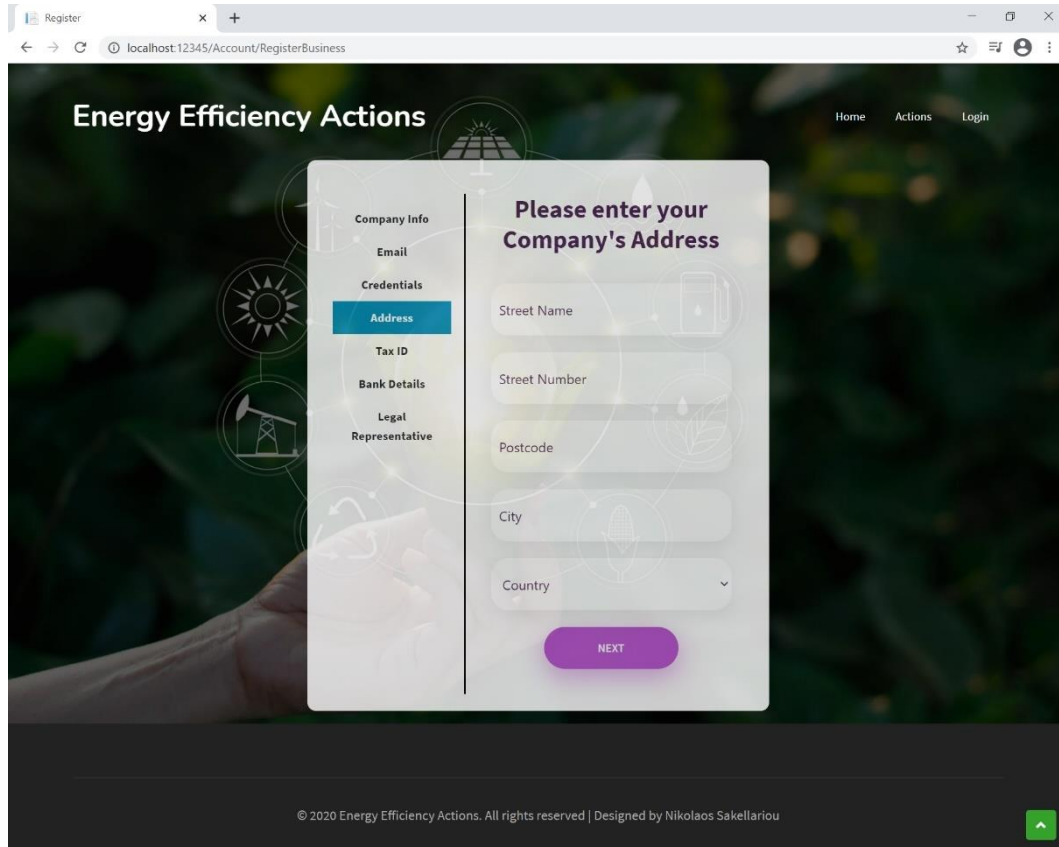
Εικόνα 12: Εγγραφή Επιχείρησης - Email



Εικόνα 13: Εγγραφή Επιχείρησης - Διαπιστευτήρια



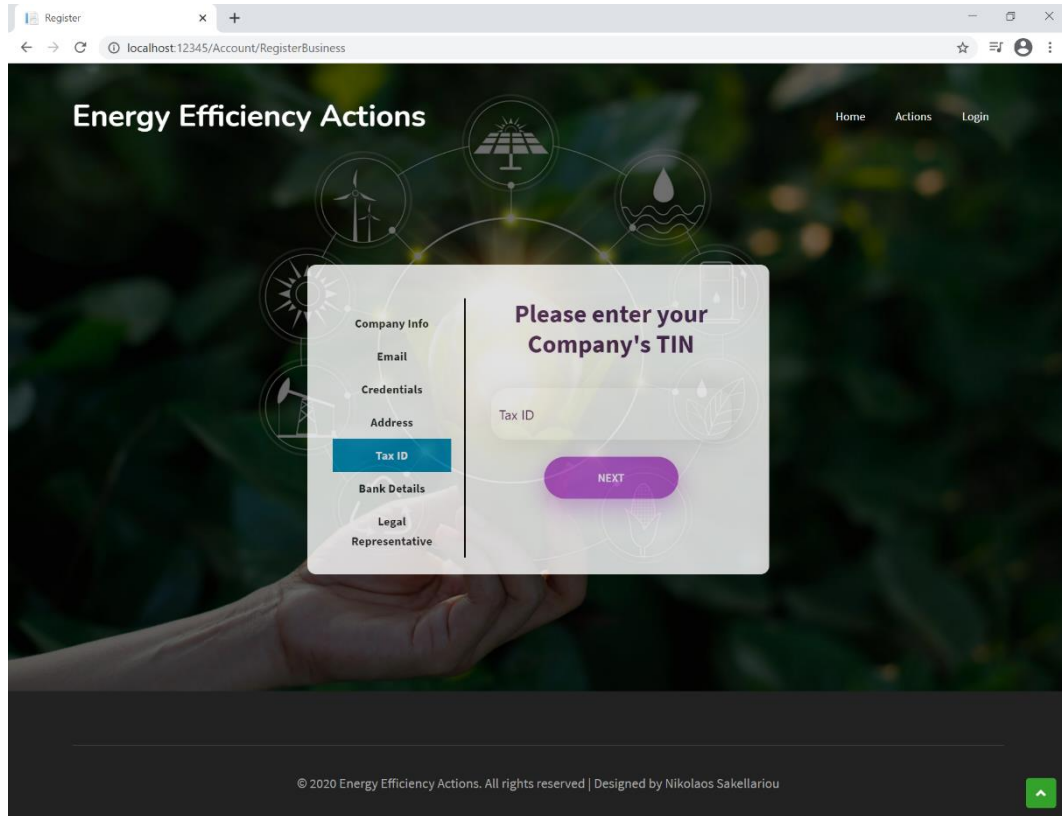
Στην τέταρτη καρτέλα, όπως παρατηρείται στην Εικόνα 13, συμπληρώνονται τα στοιχεία της διεύθυνσης της έδρας της επιχείρησης, όπως οδός, αριθμός, ταχυδρομικός κώδικας, πόλη και χώρα.



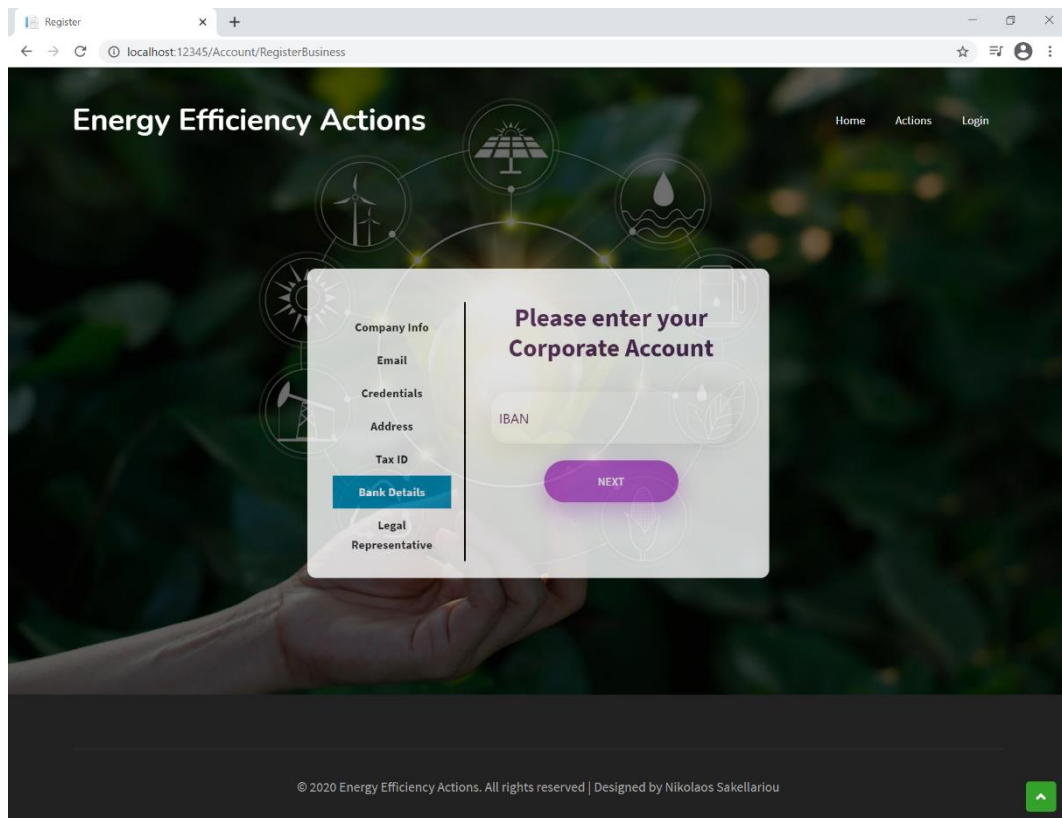
The image shows a web browser window displaying a registration form for 'Energy Efficiency Actions'. The browser's address bar shows 'localhost:12345/Account/RegisterBusiness'. The page has a dark green background with a hand holding a tablet. On the left, a vertical menu lists registration steps: Company Info, Email, Credentials, Address (highlighted in blue), Tax ID, Bank Details, Legal Representative. The main form area is titled 'Please enter your Company's Address' and contains five input fields: Street Name, Street Number, Postcode, City, and Country. A purple 'NEXT' button is at the bottom. The footer includes the copyright notice '© 2020 Energy Efficiency Actions. All rights reserved | Designed by Nikolaos Sakellariou' and a green arrow icon.

Εικόνα 14: Εγγραφή Επιχείρησης - Διεύθυνση

Στην πέμπτη καρτέλα, όπως Εικόνα 15, συμπληρώνεται ο αριθμός φορολογικού μητρώου της επιχείρησης, ενώ στην επόμενη καρτέλα ο IBAN του τραπεζικού της λογαριασμού (Εικόνα 16).

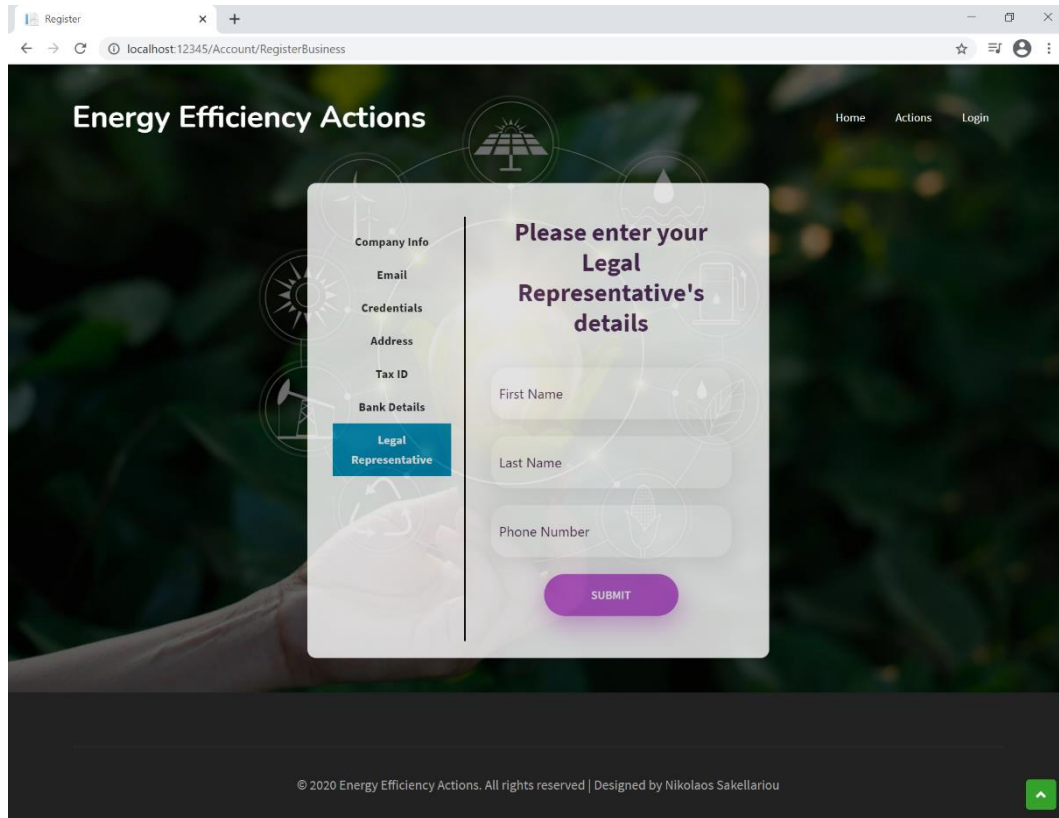


Εικόνα 15: Εγγραφή Επιχείρησης - Α.Φ.Μ



Εικόνα 16: Εγγραφή Επιχείρησης - IBAN

Τέλος, στην έβδομη καρτέλα, όπως φαίνεται στην Εικόνα 17, σημειώνονται τα στοιχεία του νόμιμου εκπροσώπου της επιχείρησης αλλά και το τηλέφωνο επικοινωνίας.



The screenshot shows a web browser window with the URL 'localhost:12345/Account/RegisterBusiness'. The page title is 'Energy Efficiency Actions' and it has navigation links for 'Home', 'Actions', and 'Login'. The main content is a registration form with a sidebar menu on the left containing: 'Company Info', 'Email', 'Credentials', 'Address', 'Tax ID', 'Bank Details', and 'Legal Representative' (which is highlighted in blue). The main form area is titled 'Please enter your Legal Representative's details' and contains three input fields: 'First Name', 'Last Name', and 'Phone Number'. A purple 'SUBMIT' button is located at the bottom of the form. The footer of the page reads '© 2020 Energy Efficiency Actions. All rights reserved | Designed by Nikolaos Sakellariou' and includes a small green arrow icon in the bottom right corner.

Εικόνα 17: Εγγραφή Επιχείρησης - Νόμιμος Εκπρόσωπος

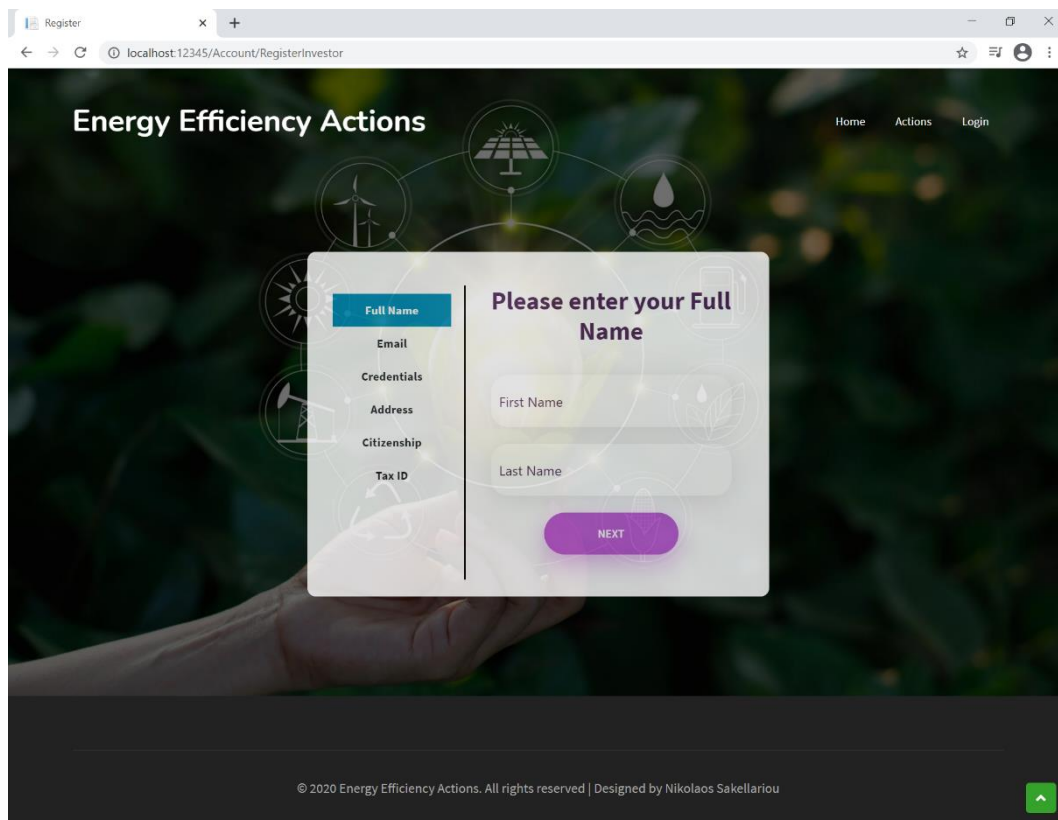
Στο τέλος της έβδομης καρτέλας βρίσκεται το κουμπί υποβολής της φόρμας εγγραφής. Κατά την υποβολή της φόρμας, για οποιοδήποτε από τα στοιχεία που έχουν εισαχθεί, εντοπιστεί ασυμφωνία με τις προδιαγραφές και τους περιορισμούς που έχουν τεθεί στο αντίστοιχο μοντέλο του View που καλείται, εμφανίζονται μέσω ενός pop-up παραθύρου τα σφάλματα που εντοπίστηκαν, προκειμένου να διορθωθούν ώστε να υποβληθεί ορθά εκ νέου η φόρμα.

#### 4.3.2. Καταχώρηση Επενδυτή

Αντίστοιχη είναι η διαδικασία καταχώρησης νέου επενδυτή. Ακολουθώντας τον σύνδεσμο του επενδυτή από την βοηθητική σελίδα εγγραφής γίνεται μεταφορά στη σελίδα

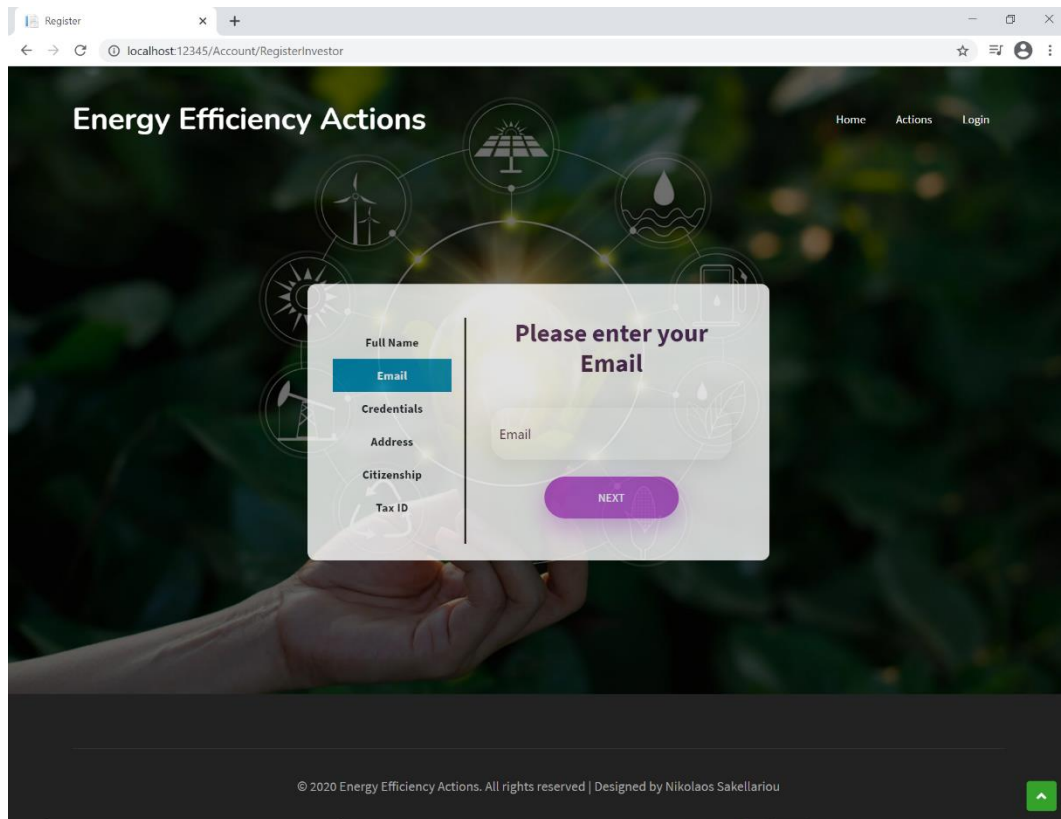
όπου θα ζητηθούν από τον επισκέπτη τα απαραίτητα στοιχεία για την ολοκλήρωση της δημιουργίας ενός λογαριασμού επενδυτή. Η φόρμα που καλείται να συμπληρώσει ο επισκέπτης είναι διαβαθμισμένη σε 6 καρτέλες, στις οποίες μπορεί να πλοηγηθεί είτε μέσω της αριστερής πλευράς της φόρμας, όπου παρουσιάζονται οι καρτέλες είτε με τα κουμπιά «Next».

Στην πρώτη καρτέλα, όπως φαίνεται στην Εικόνα 18, ο επισκέπτης συμπληρώνει το όνομα και το επώνυμό του.

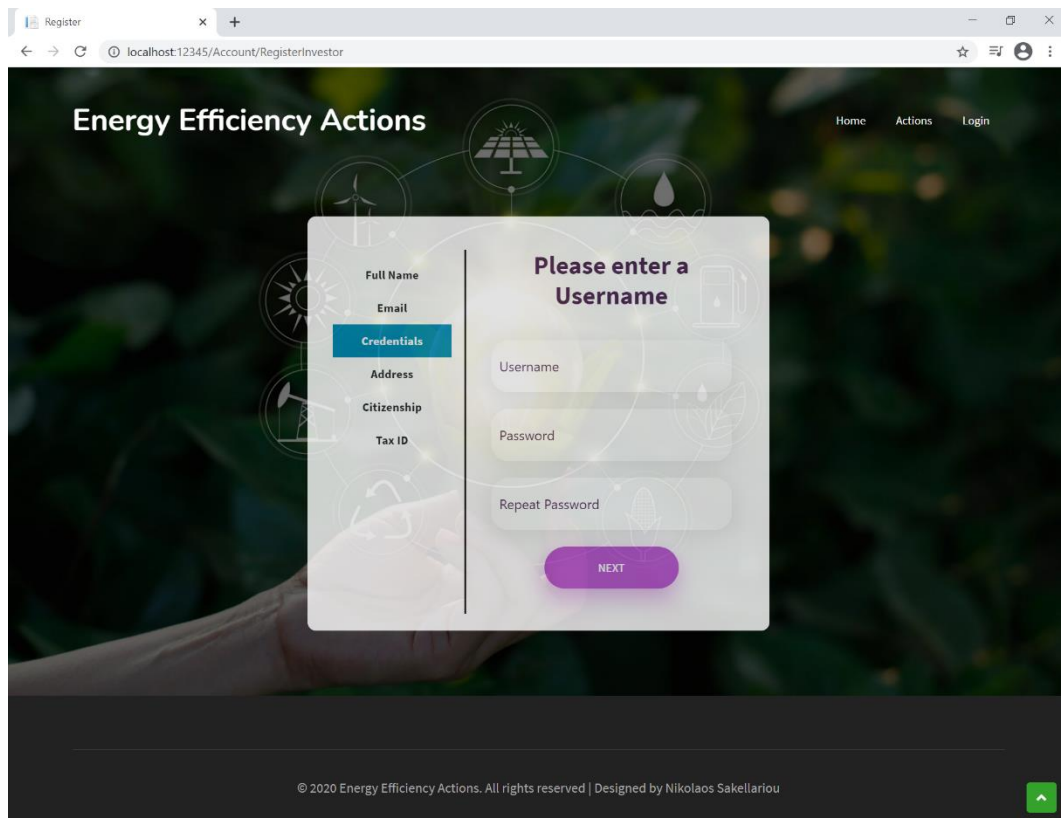


Εικόνα 18: Εγγραφή Επενδυτή - Ονοματεπώνυμο

Στη δεύτερη καρτέλα, όπως στην Εικόνα 19, απαιτείται η καταχώρηση του προσωπικού email του επισκέπτη, ενώ με την τρίτη καρτέλα, όπως αποτυπώνεται στην Εικόνα 20, συμπληρώνονται τα στοιχεία σύνδεσης του χρήστη που θα δημιουργηθεί, δηλαδή το όνομα χρήστη και ο κωδικός πρόσβασης. Το όνομα χρήστη θα πρέπει να είναι μοναδικό και ο κωδικός πρόσβασης να πληροί τις ελάχιστες απαιτήσεις που τέθηκαν με τη μέθοδο ConfigureServices της κλάσης Startup (Script 8).

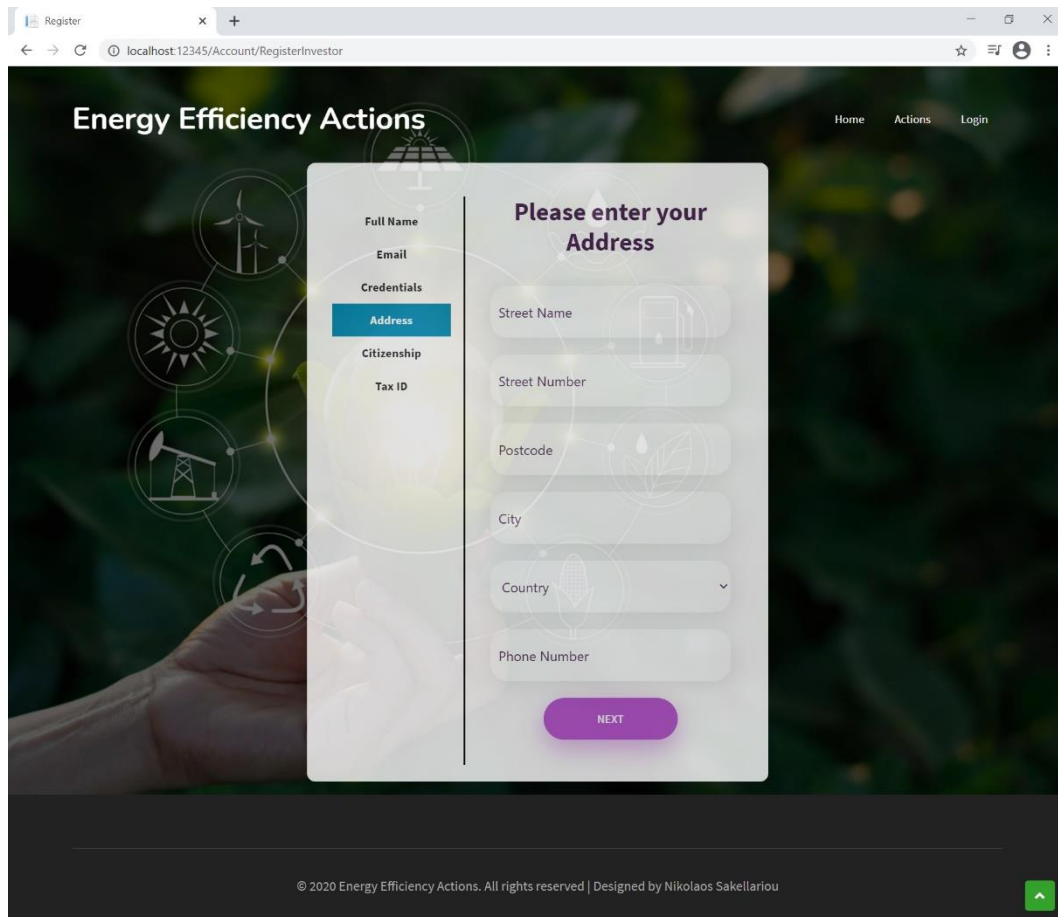


Εικόνα 19: Εγγραφή Επενδυτή - Email



Εικόνα 20: Εγγραφή Επενδυτή - Διαπιστευτήρια

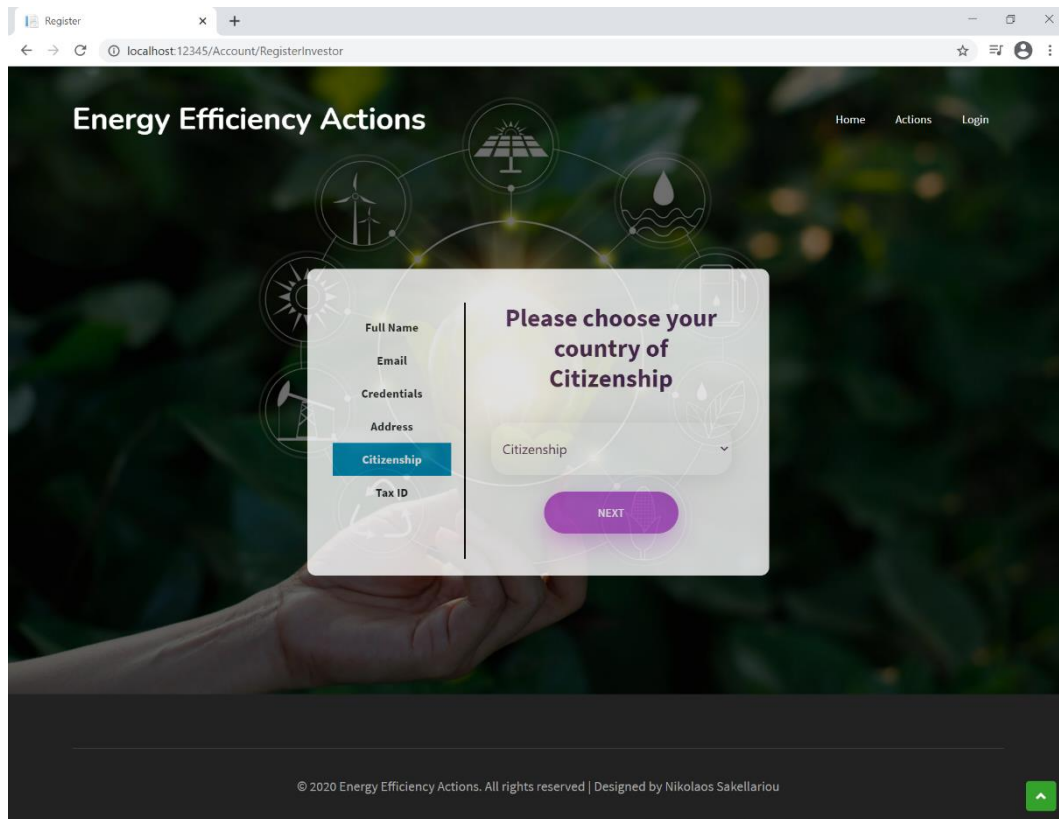
Στην τέταρτη καρτέλα, όπως στην Εικόνα 21, συμπληρώνονται τα στοιχεία της διεύθυνσης κατοικίας του επισκέπτη, όπως οδός, αριθμός, ταχυδρομικός κώδικας, πόλη, χώρα και τηλέφωνο.



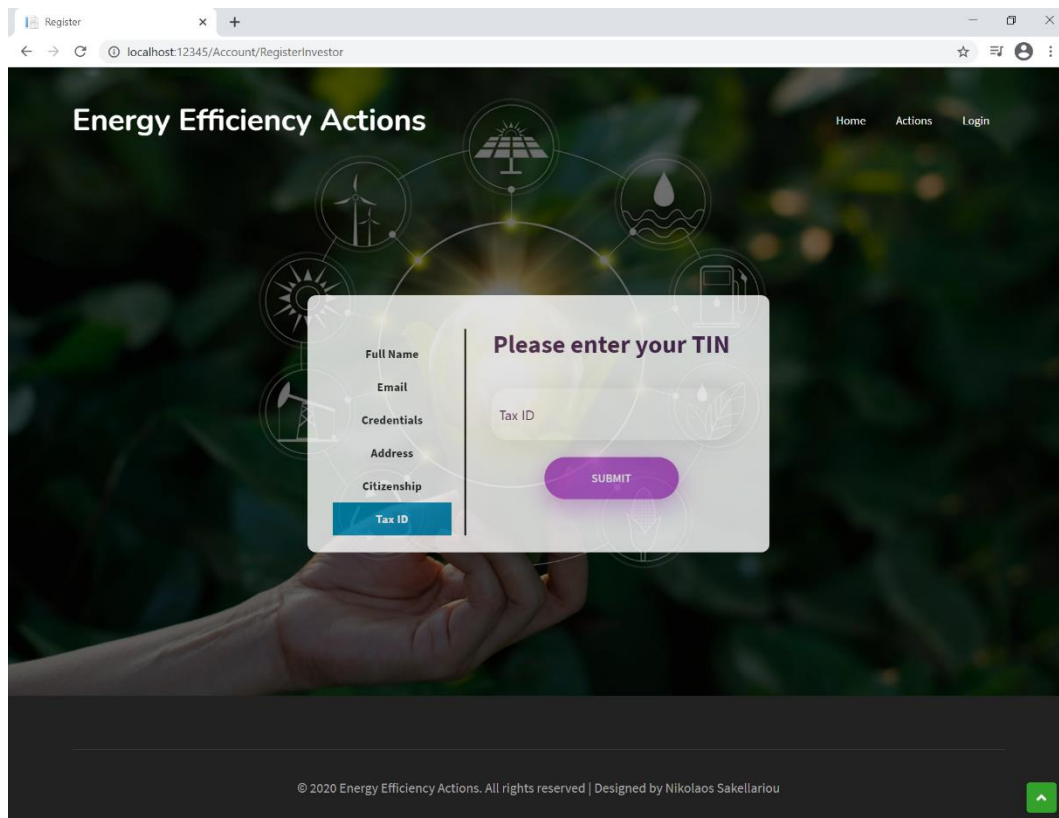
The image shows a web browser window displaying a registration form for 'Energy Efficiency Actions'. The browser's address bar shows 'localhost:12345/Account/RegisterInvestor'. The page has a dark green background with icons representing energy sources like wind turbines, solar panels, and oil rigs. A white modal form is centered on the screen, titled 'Please enter your Address'. On the left side of the modal, there is a vertical list of fields: 'Full Name', 'Email', 'Credentials', 'Address' (highlighted with a blue bar), 'Citizenship', and 'Tax ID'. The right side of the modal contains input fields for 'Street Name', 'Street Number', 'Postcode', 'City', 'Country' (a dropdown menu), and 'Phone Number'. A purple 'NEXT' button is located at the bottom of the form. At the bottom of the page, there is a copyright notice: '© 2020 Energy Efficiency Actions. All rights reserved | Designed by Nikolaos Sakellariou' and a small green arrow icon in the bottom right corner.

Εικόνα 21: Εγγραφή Επενδυτή – Διεύθυνση

Στην πέμπτη καρτέλα, όπως Εικόνα 22, επιλέγεται η υπηκοότητα του υποψήφιου επενδυτή, ενώ στην τελευταία καρτέλα συμπληρώνεται ο αριθμός φορολογικού του μητρώου (Εικόνα 23).



Εικόνα 22: Εγγραφή Επενδυτή – Υπηκοότητα



Εικόνα 23: Εγγραφή Επιχείρησης - Α.Φ.Μ

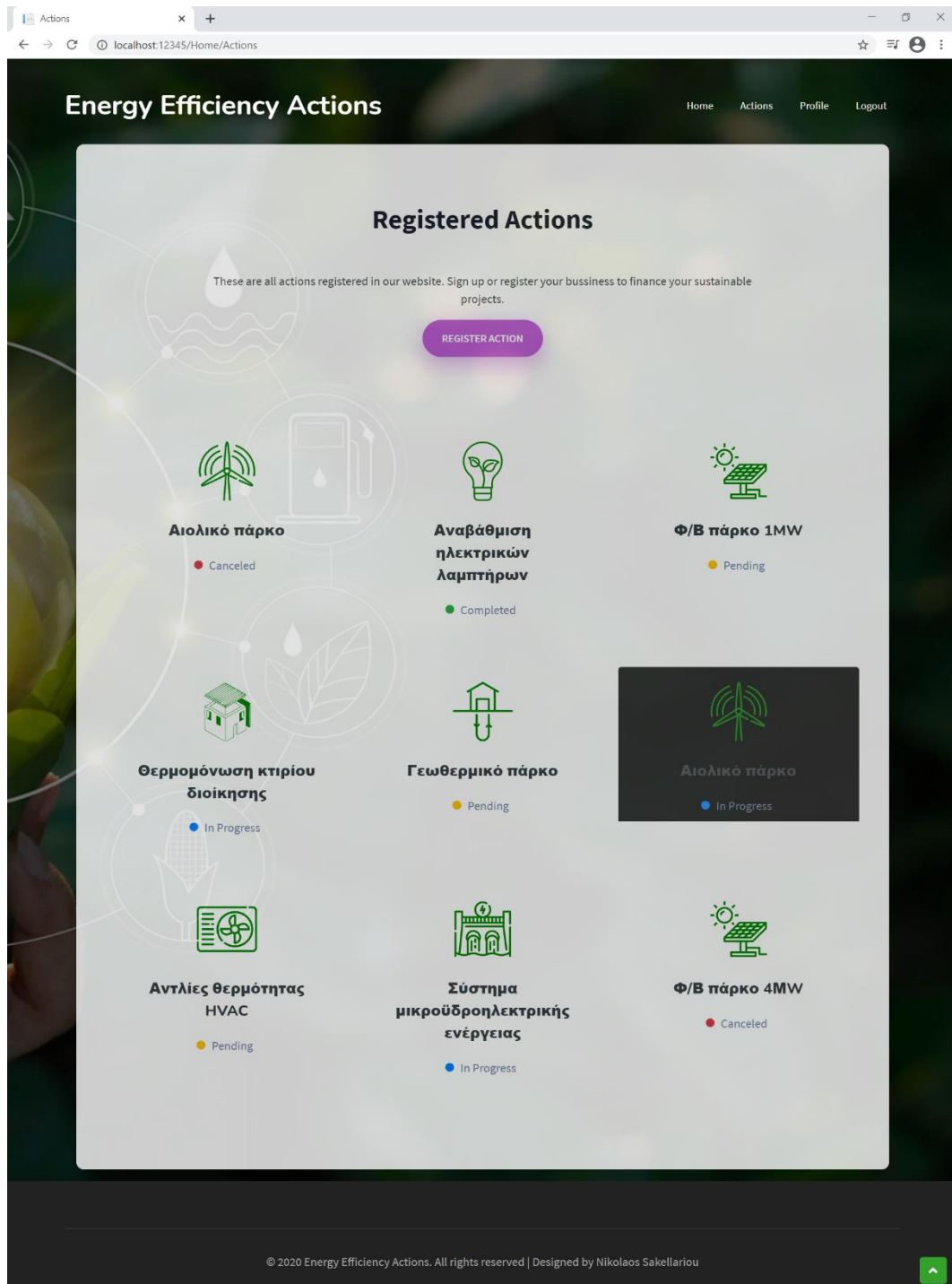
Στο τέλος της έκτης καρτέλας βρίσκεται το κουμπί υποβολής της φόρμας εγγραφής. Κατά την υποβολή της φόρμας, για οποιοδήποτε από τα στοιχεία που έχουν εισαχθεί, εντοπιστεί ασυμφωνία με τις προδιαγραφές και τους περιορισμούς που έχουν τεθεί στο αντίστοιχο μοντέλο του View που καλείται, εμφανίζονται μέσω ενός pop-up παραθύρου τα σφάλματα που εντοπίστηκαν, προκειμένου να διορθωθούν ώστε να υποβληθεί ορθά εκ νέου η φόρμα.

#### 4.4. Σελίδα καταχωρημένων δράσεων

Επιλέγοντας από το μενού πλοήγησης την εμφάνιση των δράσεων (Actions) γίνεται μεταφορά στην αντίστοιχη σελίδα (Εικόνα 24). Εδώ ο χρήστης μπορεί να δει όλες τις δράσεις που είναι καταχωρημένες στην πλατφόρμα, παρατηρώντας τον τίτλο τους, και την κατάσταση της καθεμίας εξ αυτών, ενώ κάνοντας κλικ πάνω σε μία οποιαδήποτε δράση μεταφέρεται στη σελίδα της.

Στη σελίδα αυτή μπορούν να περιηγηθούν όλοι οι επισκέπτες του ιστότοπου ανεξαρτήτως από το αν έχουν κάνει είσοδο ως χρήστες. Ωστόσο, στην περίπτωση που υπάρχει συνδεδεμένος χρήστης με λογαριασμό επιχείρησης εμφανίζεται στο επάνω μέρος της ιστοσελίδας το κουμπί «Register Action», το οποίο χρησιμοποιείται για την καταχώρηση μιας νέας δράσης, όπως παρατηρείται στην Εικόνα 24.





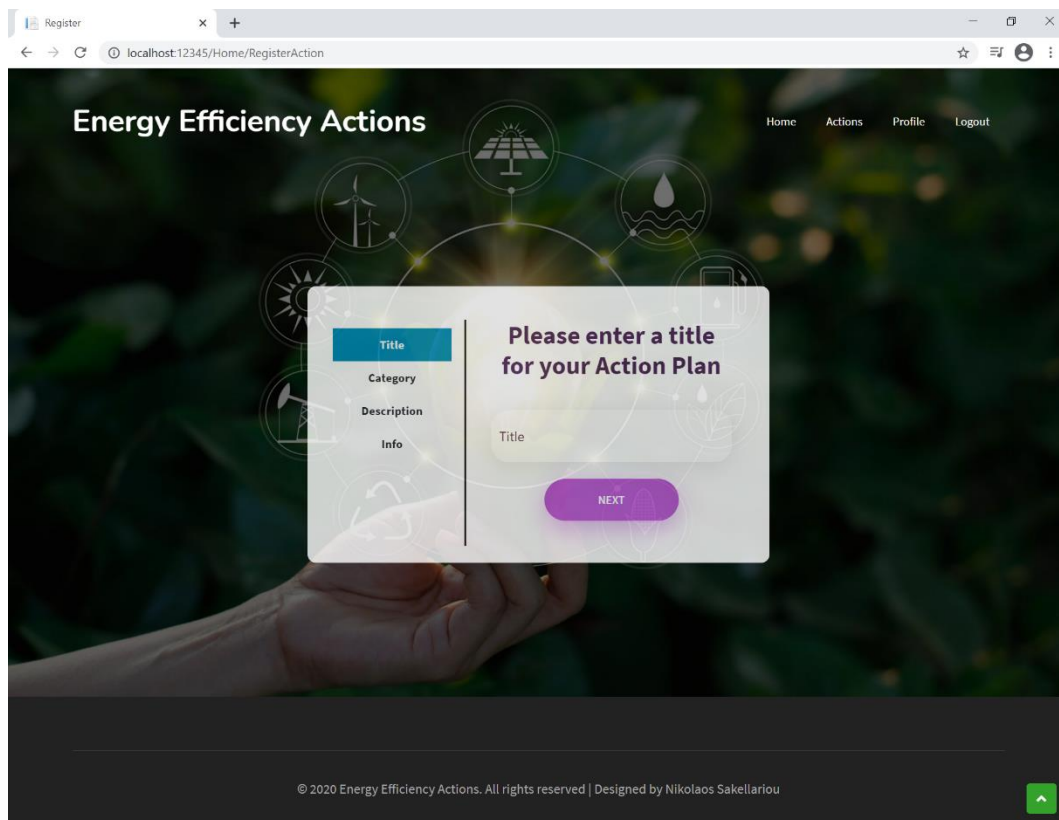
Εικόνα 24: Σελίδα καταχωρημένων δράσεων

#### 4.5. Σελίδα καταχώρησης νέας δράσης

Όπως γίνεται αντιληπτό από την προηγούμενη ενότητα θα πρέπει να υπάρχει συνδεδεμένος χρήστης με λογαριασμό επιχείρησης για να πραγματοποιηθεί η καταχώρηση

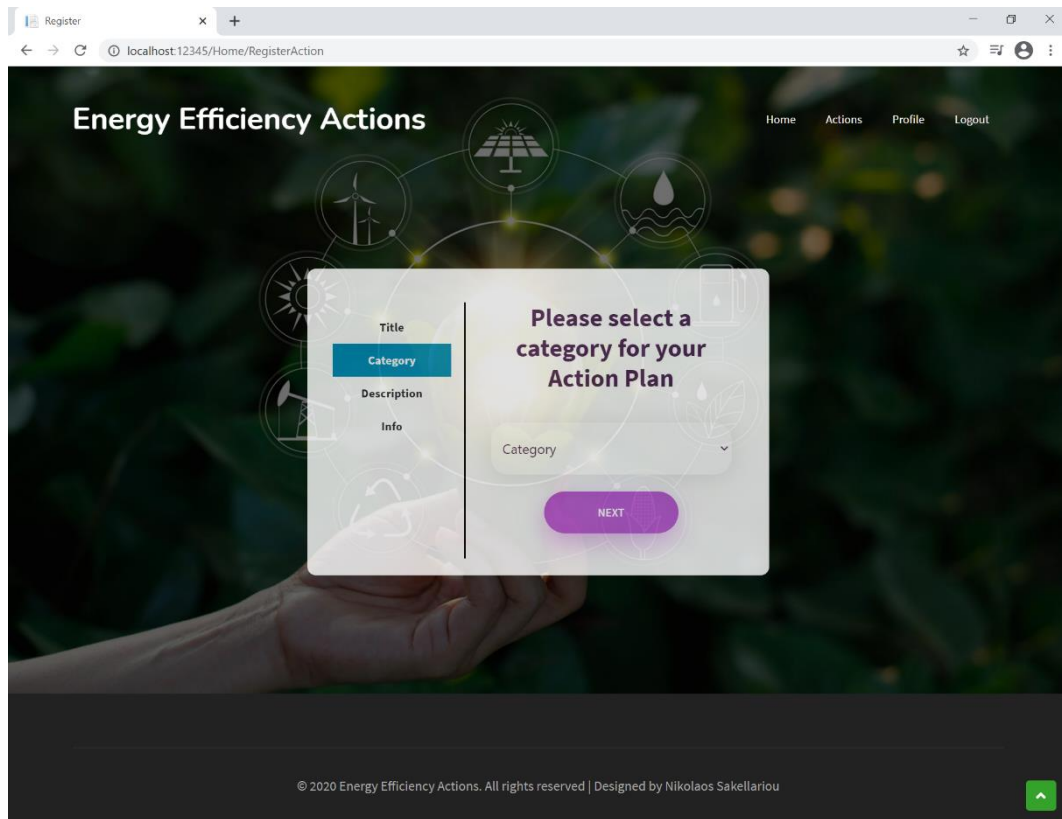
μιας νέας δράσης. Επιλέγοντας το κουμπί της εγγραφής στη σελίδα των καταχωρημένων δράσεων εκτελείται μεταφορά στη σελίδα καταχώρησης νέας δράσης, όπως φαίνεται στην Εικόνα 25.

Η φόρμα που καλείται να συμπληρώσει ο επισκέπτης είναι διαβαθμισμένη σε 4 καρτέλες, στις οποίες μπορεί να πλοηγηθεί είτε μέσω της αριστερής πλευράς της φόρμας, όπου παρουσιάζονται οι καρτέλες είτε με τα κουμπιά «Next».



Εικόνα 25: Καταχώρηση νέας δράσης - Τίτλος

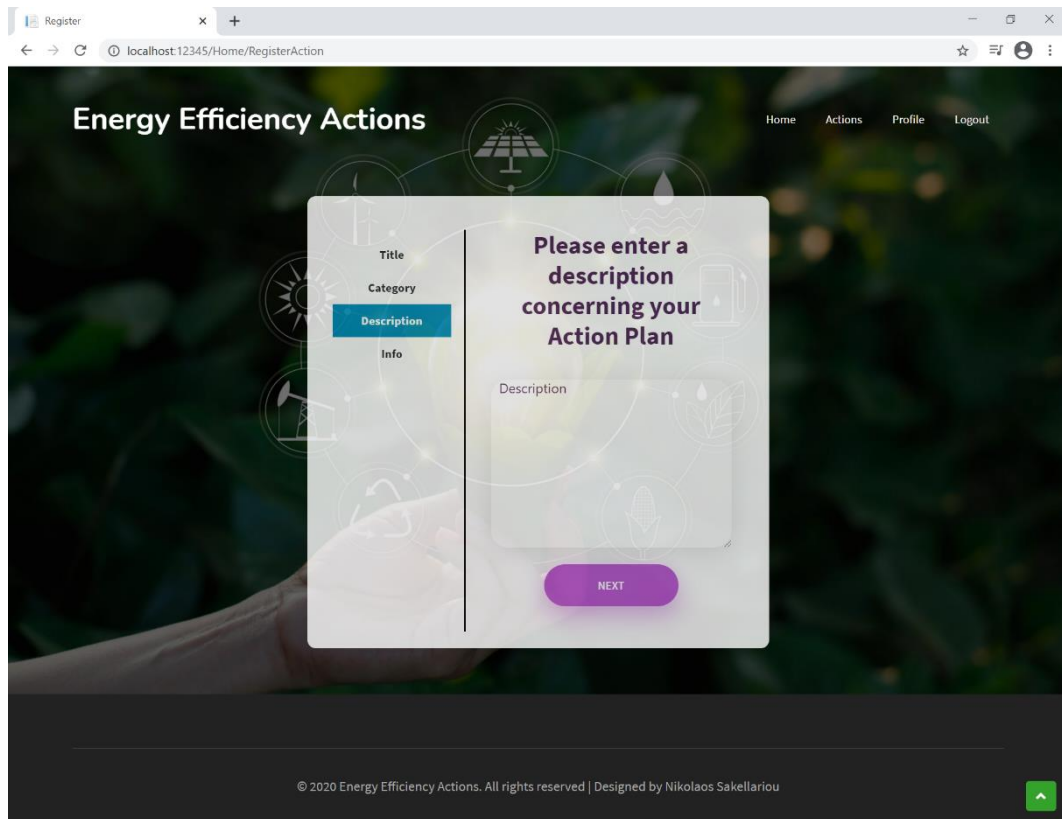
Στην πρώτη καρτέλα ζητείται η συμπλήρωση του τίτλου της δράσης που θα καταχωρηθεί, ενώ στη δεύτερη, όπως φαίνεται στην Εικόνα 26, επιλέγεται η κατηγορία της δράσης από τη λίστα.



Εικόνα 26: Καταχώρηση νέας δράσης - Κατηγορία

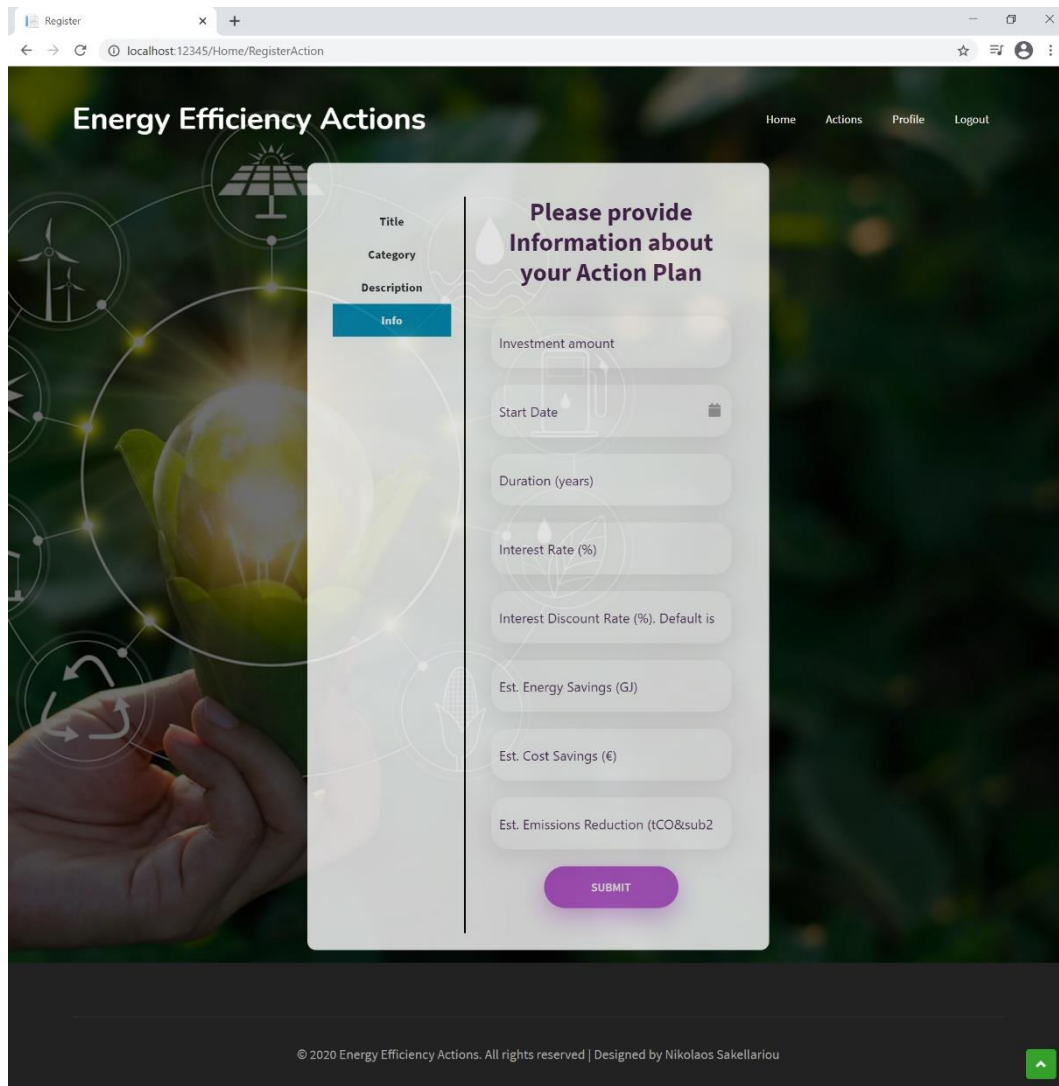
Στην τρίτη καρτέλα, όπως φαίνεται στην Εικόνα 27, ο χρήστης θα πρέπει να συμπληρώσει μία σύντομη περιγραφή της δράσης εξοικονόμησης ενέργειας που επιθυμεί να καταχωρήσει.

Τέλος, στην τέταρτη καρτέλα (Εικόνα 28) απαιτείται η συμπλήρωση όλων των χαρακτηριστικών της δράσης, όπως το απαραίτητο ποσό που απαιτείται να επενδυθεί, η ημερομηνία έναρξής της και η διάρκειά της σε έτη, το ετήσιο επιτόκιο για τους επενδυτές, το προεξοφλητικό επιτόκιο δανεισμού, αλλά και η αναμενόμενη εξοικονόμηση ενέργειας, πόρων και η αναμενόμενη μείωση σε όρους διοξειδίου του άνθρακα που θα επιφέρει η δράση.



Εικόνα 27: Καταχώρηση νέας δράσης - Περιγραφή

Στο κάτω μέρος της τελευταίας αυτής καρτέλας βρίσκεται το κουμπί υποβολής της φόρμας. Κατά την υποβολή της φόρμας, για οποιοδήποτε από τα στοιχεία που έχουν εισαχθεί, εντοπιστεί ασυμφωνία με τις προδιαγραφές και τους περιορισμούς που έχουν τεθεί στο αντίστοιχο μοντέλο του View που καλείται, εμφανίζονται μέσω ενός pop-up παραθύρου τα σφάλματα που εντοπίστηκαν, προκειμένου να διορθωθούν ώστε να υποβληθεί ορθά εκ νέου η φόρμα.

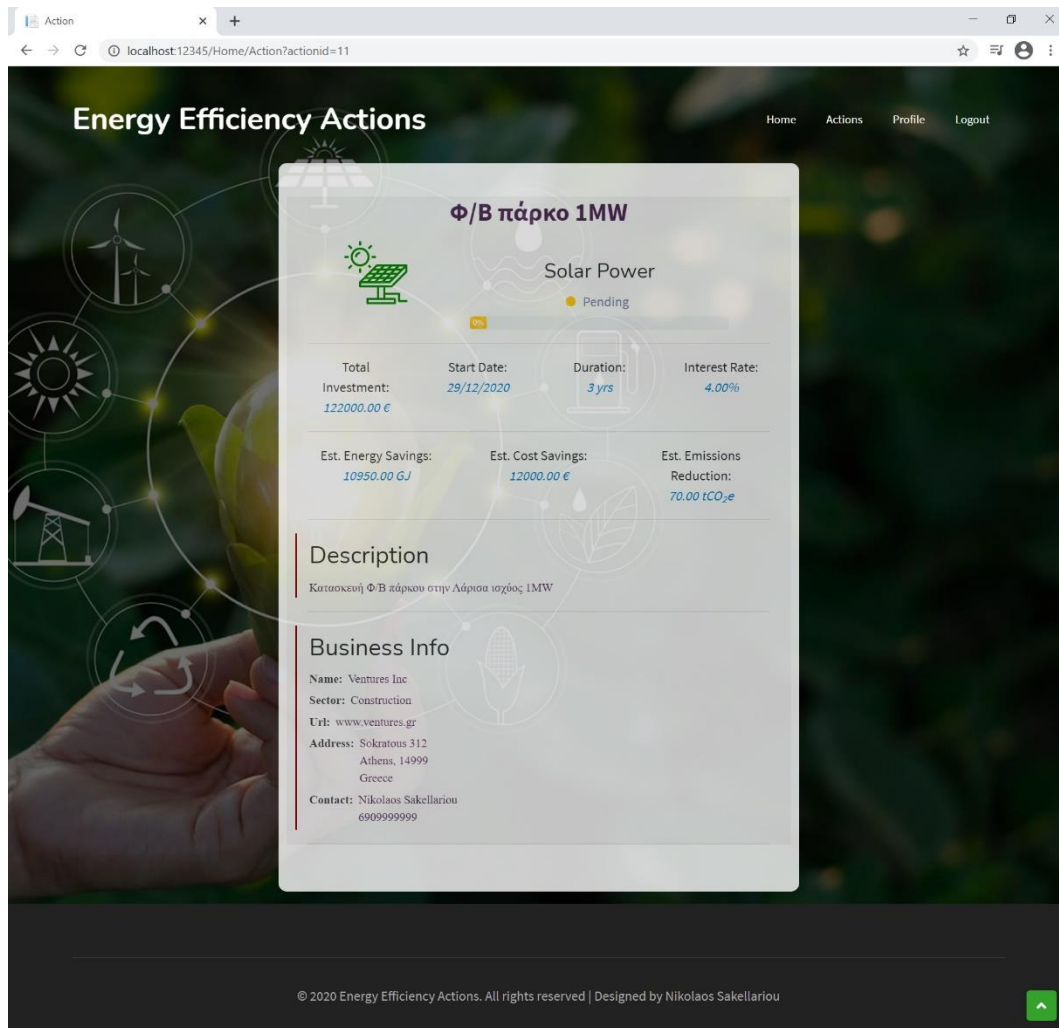


The screenshot displays a web browser window with the URL `localhost:12345/Home/RegisterAction`. The page title is "Energy Efficiency Actions". The navigation menu includes "Home", "Actions", "Profile", and "Logout". The main content area features a form titled "Please provide Information about your Action Plan". The form is divided into two columns. The left column contains a table with headers "Title", "Category", and "Description", and a blue "Info" button. The right column contains several input fields: "Investment amount", "Start Date" (with a calendar icon), "Duration (years)", "Interest Rate (%)", "Interest Discount Rate (%). Default is", "Est. Energy Savings (GJ)", "Est. Cost Savings (€)", and "Est. Emissions Reduction (tCO<sub>2</sub>)". A purple "SUBMIT" button is located at the bottom of the form. The background of the page shows a hand holding a glowing lightbulb, with various energy-related icons like a wind turbine and solar panels.

Εικόνα 28: Καταχώρηση νέας δράσης - Πληροφορίες

#### 4.6. Σελίδα ανάλυσης δράσης

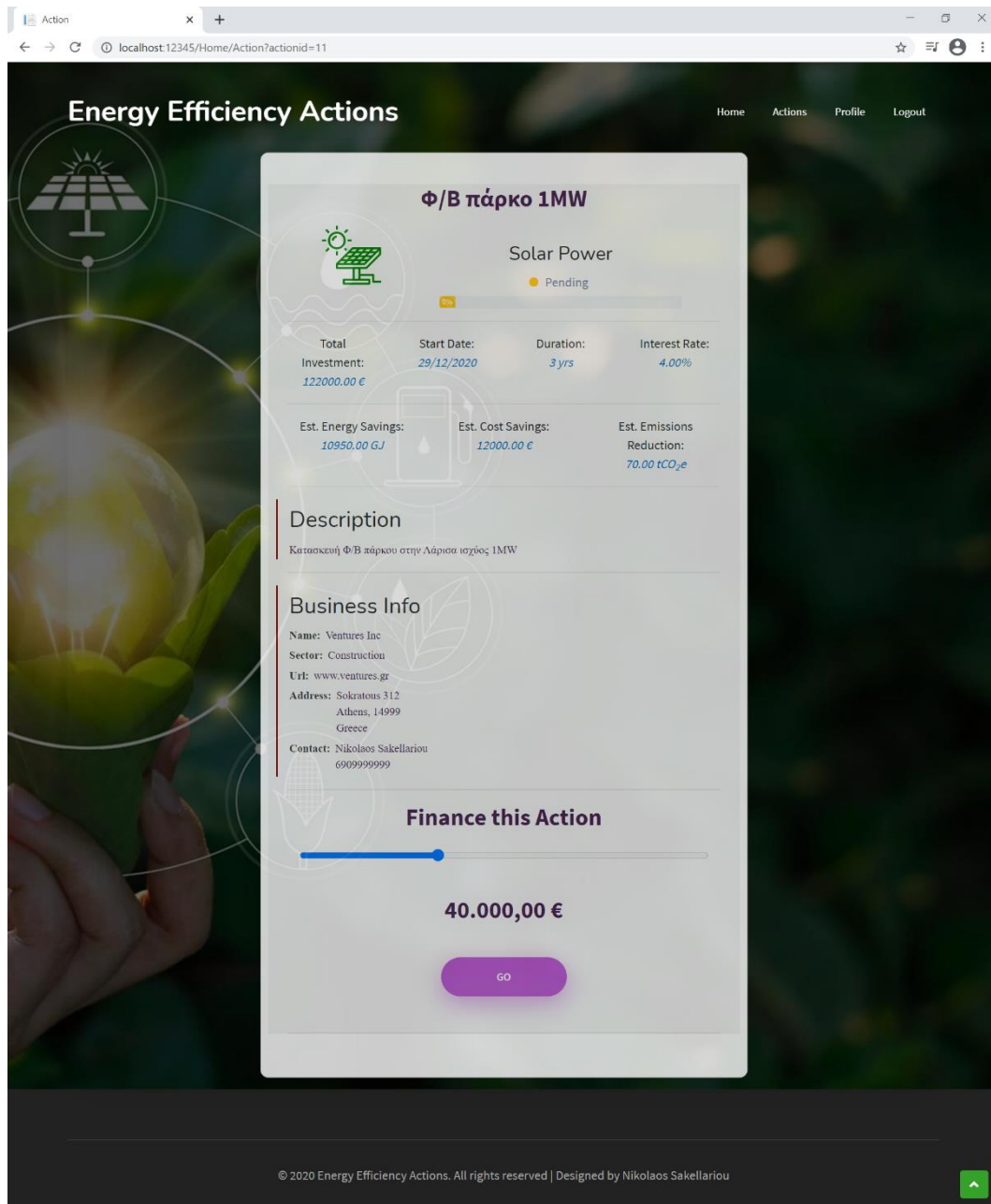
Επιλέγοντας μια δράση είτε από την αρχική σελίδα της πλατφόρμας είτε από τη σελίδα των καταχωρημένων δράσεων, εκτελείται μεταφορά στη σελίδα της Εικόνας 29, όπου παρουσιάζεται μια ακριβής ανάλυση της δράσης ως προς τα χαρακτηριστικά της. Ωστόσο, η χρησιμότητά της πέραν της ολοκληρωμένης παρουσίασης μιας δράσης στους επισκέπτες και τους χρήστες του ιστότοπου, επεκτείνεται με επιμέρους λειτουργίες για τις δύο κατηγορίες των χρηστών.



Εικόνα 29: Σελίδα δράσης – Γενική Απεικόνιση

Ειδικότερα, στην περίπτωση που ο χρήστης που επισκέπτεται τη συγκεκριμένη ιστοσελίδα αντιστοιχεί σε λογαριασμό επενδυτή και η κατάσταση της συγκεκριμένης δράσης που προβάλλεται είναι σε εξέλιξη, τότε εμφανίζεται ένα πλαίσιο στο τέλος της σελίδας, όπως παρατηρείται στην Εικόνα 30. Το πλαίσιο αυτό δίνει τη δυνατότητα στον επενδυτή να επιλέξει το ποσό με το οποίο επιθυμεί να συμμετάσχει στη χρηματοδότηση της συγκεκριμένης δράσης. Αφού επιλέξει το ποσό με τη βοήθεια ενός slider<sup>2</sup> και πιέζοντας το κουμπί «Go», εμφανίζεται ένα αναδυόμενο παράθυρο προκειμένου να επιβεβαιώσει την επιλογή του και στη συνέχεια καταχωρείται η συμμετοχή του στην χρηματοδότηση της δράσης.

<sup>2</sup> Ο slider λαμβάνει ως μέγιστη τιμή το υπολειπόμενο ποσό που απαιτείται για την υλοποίηση της δράσης και ως ελάχιστη τα 100€ εκτός και αν το υπολειπόμενο ποσό είναι μικρότερο των 100€, οπότε και λαμβάνει την τιμή 0.



Εικόνα 30: Σελίδα δράσης – Απεικόνιση Επενδυτή

Από την άλλη πλευρά, στην περίπτωση που ο συνδεδεμένος χρήστης που προβάλει την σελίδα μιας δράσης αντιστοιχεί στο λογαριασμό της επιχείρησης που έχει καταχωρήσει τη συγκεκριμένη δράση και η κατάσταση της τελευταίας είναι είτε σε εξέλιξη είτε ολοκληρωμένη, εμφανίζεται στο τέλος της σελίδας ένας συγκεντρωτικός πίνακας. Ο πίνακας αυτός, όπως αποτυπώνεται στην Εικόνα 31, παρουσιάζει τις ταμειακές ροές από και προς την επιχείρηση (εισροές - εκροές), ανά ημερομηνία και είδος.

The screenshot displays a web application interface for 'Energy Efficiency Actions'. The main content area is titled 'Θερμομόνωση κτιρίου διοίκησης' (Thermal Insulation of office building) and 'Thermal Insulation'. The status is 'In Progress'. Key financial and operational data is presented in a structured layout:

- Total Investment:** 200000.00 €
- Start Date:** 22/12/2020
- Duration:** 2 yrs
- Interest Rate:** 4.10%
- Est. Energy Savings:** 730.00 GJ
- Est. Cost Savings:** 79000.00 €
- Est. Emissions Reduction:** 76.50 tCO<sub>2</sub>e

The 'Description' section states: 'Αντικατάσταση κουφωμάτων και τοποθέτηση θερμομόνωσης στο κτίριο διοίκησης της εταιρίας.'

The 'Business Info' section provides details for 'Ventures Inc':  
Sector: Construction  
Url: www.ventures.gr  
Address: Sokratous 312, Athens, 14999, Greece  
Contact: Nikolaos Sakellariou, 6909999999

The 'Net Values' table tracks the financial performance over time:

Date	Investment	Interests	Savings	Net Amount
22/12/2020	500000.00€	0.00€	0.00€	500000€
22/12/2021	0.00€	-10000.00€	39500.00€	27595.88€
22/12/2022	-500000.00€	-10000.00€	39500.00€	-411722.14€

The final calculated value is **Action NPV: 115873.74€**.

© 2020 Energy Efficiency Actions. All rights reserved | Designed by Nikolaos Sakellariou

Εικόνα 31: Σελίδα δράσης – Απεικόνιση Επιχείρησης

Στο σημείο αυτό πρέπει να αναφερθεί ότι η τέταρτη στήλη, που παρουσιάζει την εξοικονόμηση πόρων, αντιστοιχεί στο σύνολο των πόρων που εξοικονομούνται ως αποτέλεσμα της δράσης, επιμερισμένο ισόποσα ανά έτος. Η δεύτερη και τρίτη στήλη απεικονίζουν τις ταμειακές ροές που αφορούν το κεφάλαιο της επένδυσης και την πληρωμή των τόκων αντίστοιχα. Η τελευταία στήλη αντιστοιχεί στις καθαρές παρούσες ροές βάσει του



προεξοφλητικού επιτοκίου, ενώ στο κάτω μέρος του πίνακα υπολογίζεται η ΚΠΑ της επένδυσης για την επιχείρηση. Ο τρόπος υπολογισμού της ΚΠΑ είναι απλουστευμένος για τους σκοπούς της παρούσας εργασίας.

## 4.7. Προφίλ Χρηστών

Οι χρήστες μετά τη σύνδεσή τους στην πλατφόρμα μπορούν να πλοηγηθούν στη σελίδα του προφίλ τους μέσω της επιλογής στο μενού που εμφανίζεται κατόπιν του login. Αναλόγως του τύπου χρήστη που συνδέεται, η σελίδα του προφίλ είναι τροποποιημένη ανάλογα με το αν πρόκειται για λογαριασμό επιχείρησης ή επενδυτή.

### 4.7.1. Προφίλ Επιχείρησης

Όπως παρατηρείται στην Εικόνα 32, η ιστοσελίδα του προφίλ της επιχείρησης απαρτίζεται από 3 καρτέλες. Η πλοήγηση μεταξύ αυτών πραγματοποιείται επιλέγοντας κατάλληλα από την αριστερή πλευρά της σελίδας.

Στην πρώτη καρτέλα εμφανίζονται όλα τα καταχωρημένα στοιχεία της επιχείρησης, όπως το όνομα της νομικής οντότητας, ο τομέας στον οποίο δραστηριοποιείται, το url της ιστοσελίδας της, στοιχεία διεύθυνσης, φορολογικά στοιχεία, αλλά και στοιχεία του νόμιμου εκπροσώπου της.

Energy Efficiency Actions

Home Actions Profile Logout

Account

Account Details

Legal Name

Ventures Inc

Business SectorBusiness Sector

Construction

Website url

www.ventures.gr

Street Name

Sokratous

Street Number

312

Postcode

14999

City

Athens

Country

Greece

Tax ID

123456789

IBAN

GR840000000000099990000123

Representative's First Name

Nikolaos

Representative's Last Name

Sakellariou

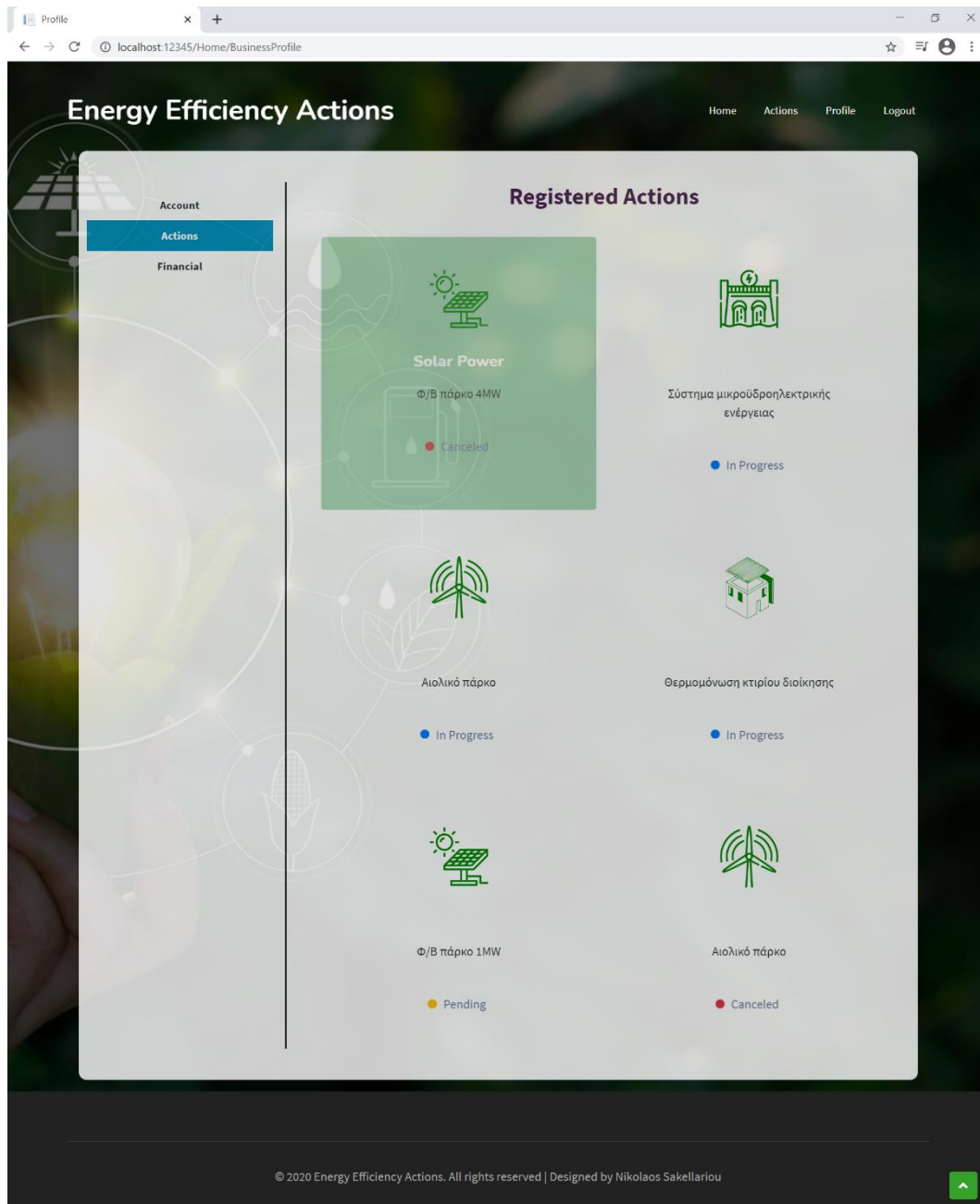
Representative's Phone Number

6909999999

© 2020 Energy Efficiency Actions. All rights reserved | Designed by Nikolaos Sakellariou

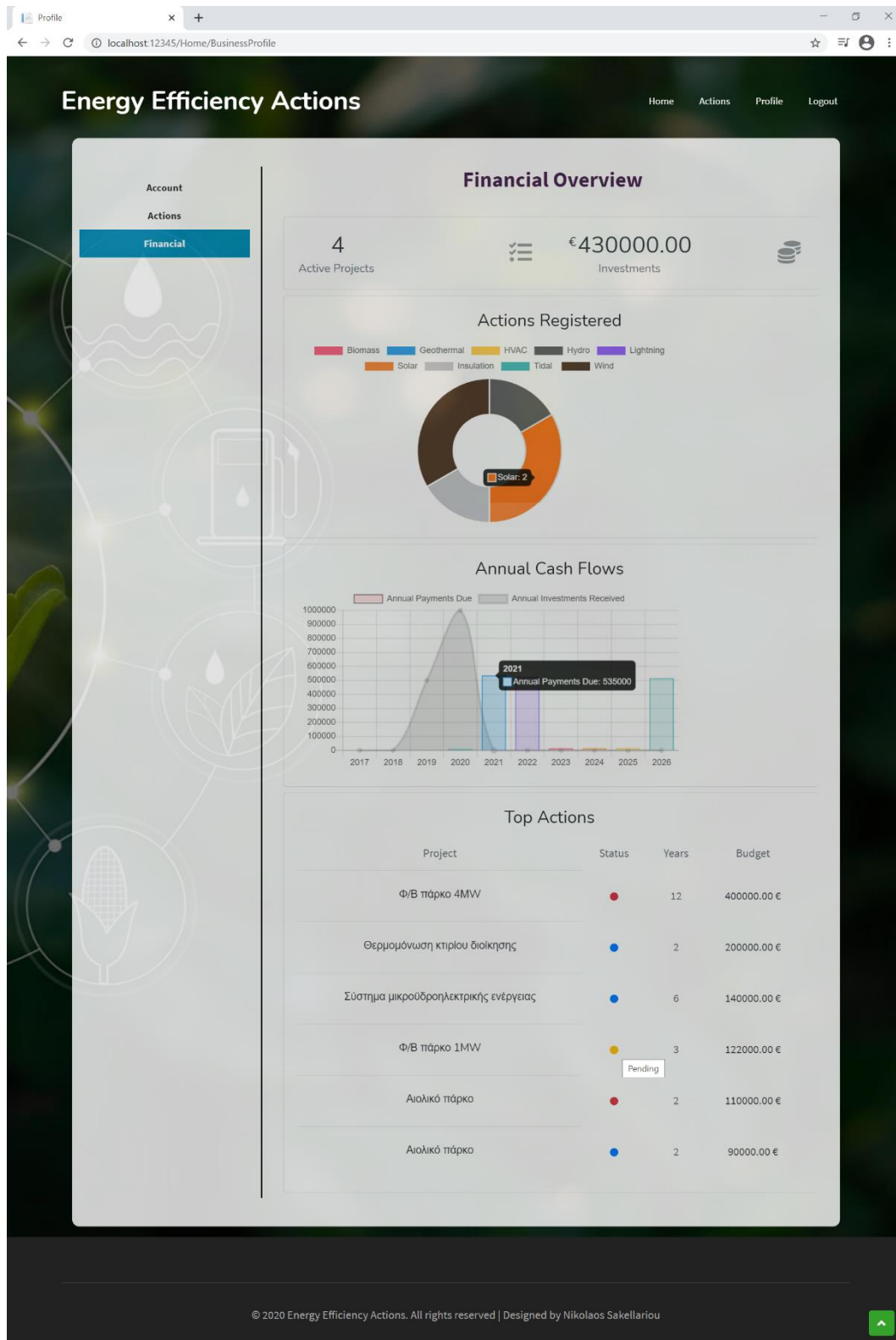
Εικόνα 32: Προφίλ Επιχείρησης – Στοιχεία Λογαριασμού

Στη δεύτερη καρτέλα, όπως φαίνεται στην Εικόνα 33, παρουσιάζονται όλες οι καταχωρημένες δράσεις της επιχείρησης, καθώς και η κατάσταση τους, ενώ επιλέγοντας μια εξ αυτών γίνεται μεταφορά στην αντίστοιχη σελίδα της δράσης.



Εικόνα 33: Προφίλ Επιχείρησης - Καταχωρημένες Δράσεις

Η τρίτη καρτέλα, της Εικόνας 34, παρέχει την οικονομική επισκόπηση της επιχείρησης με πληροφορίες σχετικά με τις δράσεις και τα οικονομικά αποτελέσματα αυτών.



Εικόνα 34: Προφίλ Επιχείρησης – Οικονομική Επισκόπηση

Ειδικότερα, παρέχονται πληροφορίες για τις δράσεις εξοικονόμησης ενέργειας που έχει καταχωρήσει η επιχείρηση και είναι σε αναμονή ή εξέλιξη (Active Projects) και τις συνολικές

επενδύσεις που έχει συγκεντρώσει από τις εν εξελίξει και τις ολοκληρωμένες δράσεις (Investments). Επιπρόσθετα, μέσω ενός διαδραστικού διαγράμματος τύπου πίτας διακρίνεται η διάρθρωση του συνόλου των καταχωρημένων δράσεων της ανά κατηγορία. Το επόμενο διάγραμμα εμφανίζει τις ετήσιες ταμειακές ροές της επιχείρησης που αφορούν είτε σε εισροές κεφαλαίου είτε σε εκροές σε τόκους και επιστροφές αρχικού κεφαλαίου στους επενδυτές. Τέλος, ο πίνακας στο τέλος της καρτέλας δείχνει σε φθίνουσα ταξινόμηση ανά ποσό επένδυσης το σύνολο των καταχωρημένων δράσεων της επιχείρησης.

#### *4.7.2. Προφίλ Επενδυτή*

Αντίστοιχα, όπως γίνεται αντιληπτό στην Εικόνα 35, αρθρώνεται η ιστοσελίδα του προφίλ του επενδυτή, η οποία απαρτίζεται επίσης από 3 καρτέλες. Η πλοήγηση μεταξύ αυτών των καρτελών πραγματοποιείται επιλέγοντας κατάλληλα από την αριστερή πλευρά της σελίδας.

Στην πρώτη καρτέλα, της Εικόνας 34, εμφανίζονται όλα τα καταχωρημένα στοιχεία του επενδυτή, όπως το ονοματεπώνυμο, τα στοιχεία διεύθυνσής του καθώς και τα φορολογικά του στοιχεία.

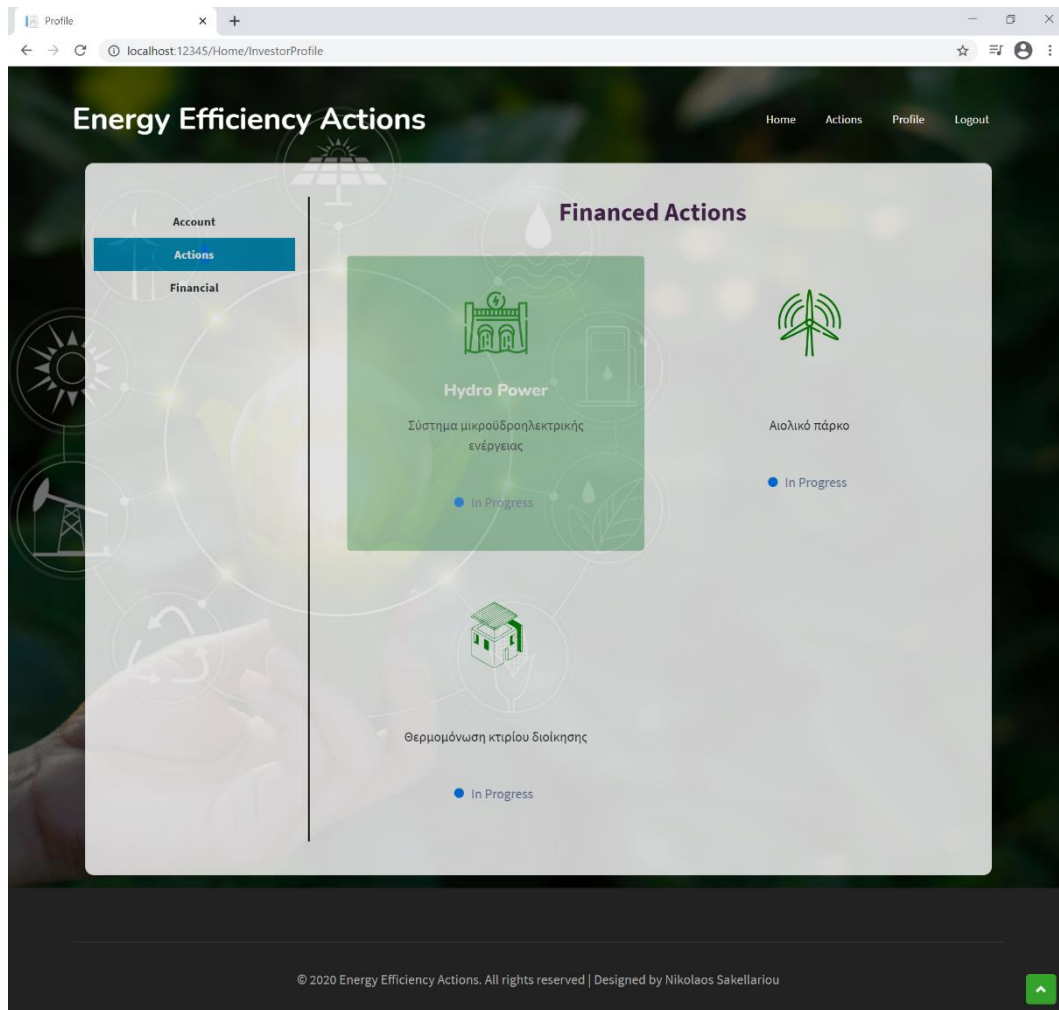
The screenshot displays a web browser window with the URL `localhost:12345/Home/InvestorProfile`. The page title is "Energy Efficiency Actions" and the navigation menu includes "Home", "Actions", "Profile", and "Logout". The main content area is titled "Account Details" and features a sidebar with "Account", "Actions", and "Financial" options. The form fields are as follows:

Field Label	Value
First Name	Νικόλαος
Last Name	Σακελλαρίου
Street Name	Πανεπιστημίου
Street Number	368
Postcode	11999
City	Αθήνα
Phone Number	6909999999
Country	Greece
Citizenship	Greece
Tax ID	123131231

At the bottom of the page, there is a copyright notice: "© 2020 Energy Efficiency Actions. All rights reserved | Designed by Nikolaos Sakellariou" and a small green arrow icon in the bottom right corner.

Εικόνα 35: Προφίλ Επενδυτή - Στοιχεία Λογαριασμού

Στη δεύτερη καρτέλα, όπως φαίνεται στην Εικόνα 36, παρουσιάζονται όλες οι δράσεις στις οποίες μετέχει ο επενδυτής, καθώς και η κατάσταση τους, ενώ επιλέγοντας μια εξ αυτών εκτελείται μεταφορά στην αντίστοιχη σελίδα της δράσης.

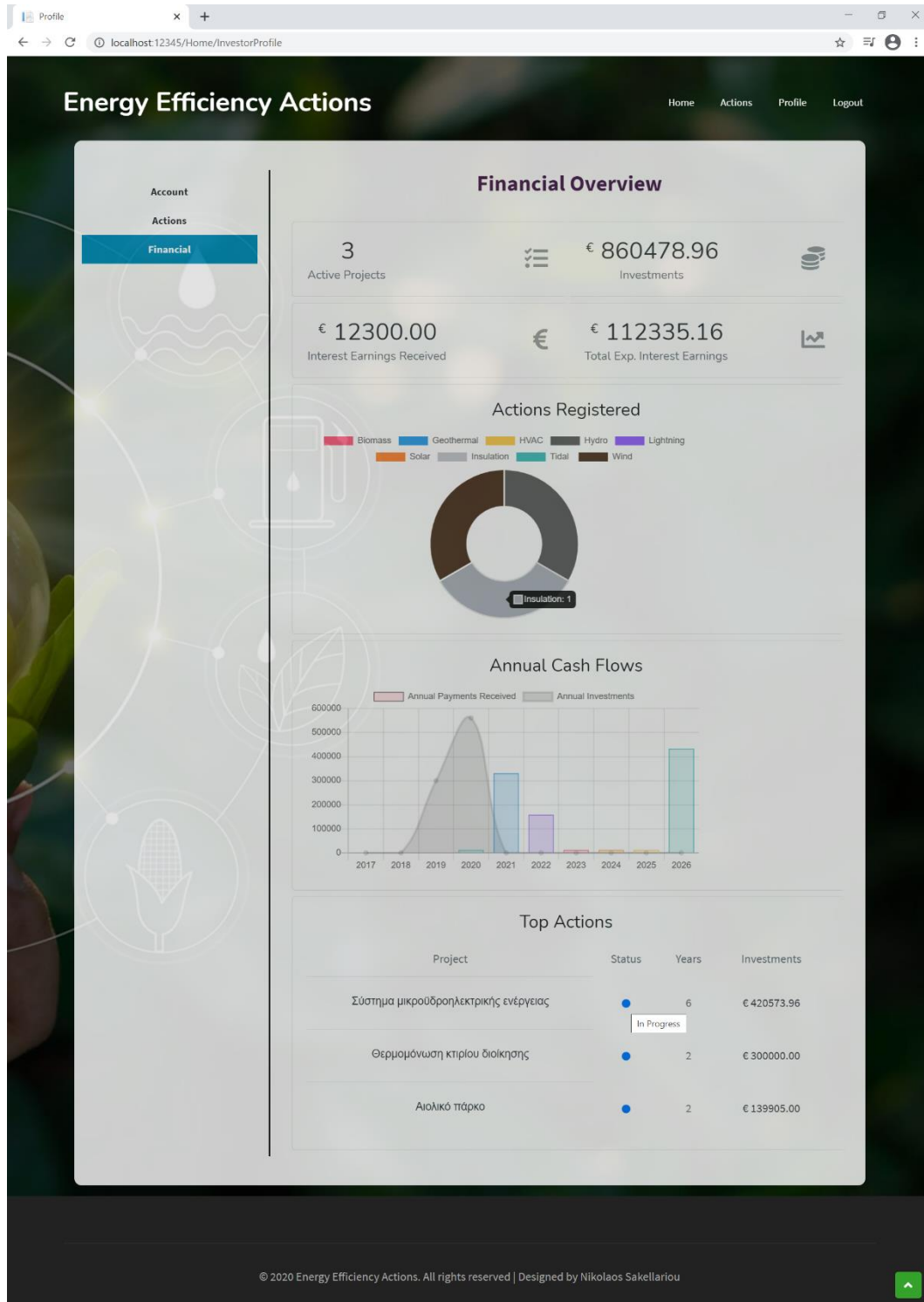


Εικόνα 36: Προφίλ Επενδυτή – Συμμετοχές σε Δράσεις

Η τρίτη καρτέλα, της Εικόνας 37, παρέχει την οικονομική επισκόπηση του προφίλ του επενδυτή με πληροφορίες σχετικά με τις δράσεις στις οποίες συμμετέχει και τα οικονομικά αποτελέσματα των επενδύσεών του.

Ειδικότερα, παρέχονται πληροφορίες για τις δράσεις εξοικονόμησης ενέργειας στις οποίες συμμετέχει και βρίσκονται σε αναμονή ή εξέλιξη (Active Projects), τις συνολικές επενδύσεις στις οποίες έχει προχωρήσει για τις εν εξελίξει και τις ολοκληρωμένες δράσεις (Investments), τα έσοδα από τόκους που έχει λάβει (Interest Earnings Received) αλλά και το σύνολο των εσόδων σε τόκους που προκύπτουν από τις επενδύσεις του (Total Exp. Interest Earnings). Επιπρόσθετα, μέσω ενός διαδραστικού διαγράμματος τύπου πίτας διακρίνεται η διάρθρωση του συνόλου των δράσεων στις οποίες συμμετέχει ανά κατηγορία, ενώ το επόμενο διάγραμμα εμφανίζει τις ετήσιες ταμειακές ροές για τον επενδυτή που αφορούν είτε σε εισροές σε τόκους και επιστροφές αρχικού κεφαλαίου είτε σε εκροές κεφαλαίου προς τις επιχειρήσεις.

Τέλος, ο πίνακας στο τέλος της καρτέλας δείχνει σε φθίνουσα ταξινόμηση ανά ποσό συμμετοχής τις δράσεις στις οποίες έχει επενδύσει.



Εικόνα 37: Προφίλ Επενδυτή – Οικονομική Επισκόπηση



## 5. ΣΥΜΠΕΡΑΣΜΑΤΑ

Οι συνεχώς αυξανόμενες ανάγκες σε ενεργειακούς πόρους και η διαρκής επιβάρυνση του περιβάλλοντος ως αποτέλεσμα της χρήσης ορυκτών καυσίμων για την παραγωγή ενέργειας σε συνδυασμό με μια πληθώρα άλλων παραγόντων έχει οδηγήσει στην στροφή προς τη χρήση πιο ήπιων και καθαρών μορφών ενέργειας αλλά και στην προσπάθεια μείωσης της καταναλισκόμενης ενέργειας με τη βοήθεια πιο σύγχρονων και αποδοτικών μεθόδων.

Η έννοια της ενεργειακής ασφάλειας αποτελεί εξέχουσα σημασίας ζήτημα για την ΕΕ, δεδομένου ότι η τελευταία δε διαθέτει επαρκείς εγχώριες πηγές ενέργειας. Δίχως τα αναγκαία γεωπολιτικά μέσα, η ΕΕ είναι αναγκασμένη να εισάγει ενεργειακούς πόρους έτσι ώστε να καλύψει τις ανάγκες της. Έχοντας κατανοήσει τη βαρύνουσα σημασία του ενεργειακού ζητήματος εμπλουτίζει συνεχώς την ενεργειακή της στρατηγική με νέους θεσμούς, μηχανισμούς, προγράμματα και μελλοντικά σχέδια, που σκοπό έχουν να διασφαλίσουν τον ενεργειακό εφοδιασμό των κρατών-μελών της. Ειδικότερα, έχει θέσει συγκεκριμένους μετρήσιμους στόχους για τις επόμενες δεκαετίες όσον αφορά τόσο την ένταση της ενεργειακής απόδοσης όσο και την αύξηση του μεριδίου των ΑΠΕ στο ενεργειακό της μείγμα.

Καθίσταται, λοιπόν, σαφές ότι η παρούσα εργασία κινείται στα πλαίσια της επίτευξης των παραπάνω στόχων και στην προσπάθεια δημιουργίας ενός εργαλείου ικανού να υποστηρίξει την προώθηση νέων δράσεων εξοικονόμησης ενέργειας. Μέσω της προσπάθειας αυτής ενισχύεται η ενεργειακή ασφάλεια τόσο σε εθνικό όσο και ευρωπαϊκό επίπεδο, καθώς μειώνονται οι ενεργειακές απαιτήσεις και κατά συνέπεια οι ανάγκες για εισαγωγές ενεργειακών πόρων, μετριάζοντας το ζήτημα του ενεργειακού εφοδιασμού της ΕΕ.

Στα πλαίσια αυτά, η χρήση της εφαρμογής συμβάλει μακροπρόθεσμα στην διαμόρφωση μιας «πράσινης» και «καθαρής» οικονομίας με σκοπό την βιωσιμότητα και την ανάπτυξη σε ένα απαλλαγμένο από ρύπους μέλλον.

Η πλατφόρμα που έχει αναπτυχθεί απευθύνεται κυρίως σε επιχειρήσεις που επιθυμούν να αναλάβουν τέτοιες δράσεις και αναζητούν την χρηματοδότηση από επενδυτές. Φέρνει σε επαφή τα δύο μέρη με έναν απλό και εύκολο τρόπο και δίνει τη δυνατότητα στους επενδυτές να επιλέξουν τη δράση την οποία επιθυμούν να χρηματοδοτήσουν και το ποσό με το οποίο θα συμμετάσχουν. Η ευκολία που προσφέρει η πλατφόρμα στην αναζήτηση ιδιωτικών κεφαλαίων για τη χρηματοδότηση έργων εξοικονόμησης ενέργειας μπορεί να οδηγήσει σε αύξηση

ανάληψης τέτοιων δράσεων με στόχο την επίτευξη των ευρωπαϊκών και εθνικών ενεργειακών στρατηγικών, ενισχύοντας παράλληλα την οικονομία.

Τα οφέλη που προσφέρει η πλατφόρμα αναφέρονται όχι μόνο στις επιχειρήσεις αλλά και στους επενδυτές. Για τις ίδιες τις επιχειρήσεις προσφέρεται η δυνατότητα αναζήτησης κεφαλαίου με μικρό κόστος δανεισμού για την ανάληψη δράσεων οι οποίες θα επιφέρουν μακροπρόθεσμα μείωση του κόστους παραγωγής ή λειτουργίας. Από την πλευρά τους οι επενδυτές μπορούν να εξασφαλίσουν ένα επιτόκιο σημαντικά υψηλότερο από αυτό που μπορεί να προσφέρεται από τα χρηματοπιστωτικά ιδρύματα.

Επιπλέον, ένα χαρακτηριστικό αυτής της εφαρμογής που ξεχωρίζει είναι η αυτοματοποίηση ορισμένων διαδικασιών με τέτοιον τρόπο ώστε να μην απαιτούνται οποιεσδήποτε ενέργειες από το διαχειριστή για τη λειτουργία της. Συγκεκριμένα, η υλοποίηση της πλατφόρμας έχει σχεδιαστεί έτσι ώστε να μην απαιτείται η παρουσία ενός διαχειριστή συστήματος για την επίβλεψη, το συντονισμό και το χειρισμό οποιασδήποτε λειτουργίας της.

Ακόμη, αξίζει να σημειωθεί ότι, η εφαρμογή αποτελεί παράλληλα ένα βοήθημα για τη χρηματοοικονομική αποτύπωση και πληροφόρηση των ενδιαφερόμενων μερών αναφορικά με τις δράσεις εξοικονόμησης ενέργειας. Συνεπώς, δύναται να αποτελέσει ένα σημαντικό εργαλείο για τις διοικήσεις των επιχειρήσεων που στοχεύουν στην υλοποίηση τέτοιων δράσεων.

Τέλος, μία ακόμη λειτουργία της εφαρμογής αποτελεί η συλλογή δεδομένων, όπως οι πληροφορίες που αποθηκεύονται στη βάση δεδομένων της και οι οποίες μπορούν να αξιοποιηθούν για την περαιτέρω ανάλυση των δράσεων.

Αν και η εφαρμογή έχει αναπτυχθεί για εκπαιδευτικούς σκοπούς και συνεπώς διαθέτει περιορισμένες δυνατότητες, δύναται ωστόσο να επεκταθεί έτσι ώστε να καλύπτει ικανά τις ανάγκες μια πλήρους λειτουργικής εφαρμογής. Παράλληλα, η επέκτασή της μπορεί να λάβει χώρα και στον τρόπο λειτουργίας της αφού θα μπορούσε να αποτελέσει ένα σημαντικό εργαλείο για την υποβοήθηση δράσεων εξοικονόμησης ενέργειας στο δημόσιο τομέα.

## 6. ΒΙΒΛΙΟΓΡΑΦΙΚΕΣ ΑΝΑΦΟΡΕΣ

- [1] «Μια Ευρωπαϊκή Πράσινη Συμφωνία,»  
Available: [https://ec.europa.eu/info/strategy/priorities-2019-2024/european-green-deal\\_el#-](https://ec.europa.eu/info/strategy/priorities-2019-2024/european-green-deal_el#-).
- [2] «Ενεργειακή πολιτική: Γενικές Αρχές | Θεματολογικά δελτία για την Ευρωπαϊκή Ένωση | Ευρωπαϊκό Κοινοβούλιο,»  
Available: <https://www.europarl.europa.eu/factsheets/el/sheet/68/ενεργειακη-πολιτικη-γενικες-αρχες>.
- [3] «Visual Studio 2019,»  
Available: <https://visualstudio.microsoft.com/vs>.
- [4] «What is .NET Framework? A software development framework,»  
Available: <https://dotnet.microsoft.com/learn/dotnet/what-is-dotnet-framework>.
- [5] «Get started with .NET Framework,»  
Available: <https://docs.microsoft.com/el-gr/dotnet/framework/get-started>.
- [6] «What is .NET Framework? A software development framework,»  
Available: <https://dotnet.microsoft.com/learn/dotnet/what-is-dotnet-framework>.
- [7] «Language-Integrated Query (LINQ) (C#),»  
Available: <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/linq>.
- [8] «What is ASP.NET? | .NET,»  
Available: <https://dotnet.microsoft.com/learn/aspnet/what-is-aspnet>.
- [9] «What is ASP.NET Core? A cross-platform web-development framework,»  
Available: <https://dotnet.microsoft.com/learn/aspnet/what-is-aspnet-core>.
- [10] «Overview of ASP.NET Core MVC,»  
Available: <https://docs.microsoft.com/en-us/aspnet/core/mvc/overview?view=aspnetcore-3.1>.
- [11] «Introduction to Identity - ASP.NET documentation,»  
Available: <https://aspnetcore.readthedocs.io/en/stable/security/authentication/identity.html>.
- [12] E. Balagurusamy, «Programming in C#: A Primer,» New Delhi, McGraw-Hill, 2009, p. 8.
- [13] «Javascript - Overview - Tutorialspoint,»  
Available: [https://www.tutorialspoint.com/javascript/javascript\\_overview.htm](https://www.tutorialspoint.com/javascript/javascript_overview.htm).
- [14] «jQuery,»  
Available: <https://jquery.com>.

- [15] «Ajax | jQuery Learning Center,»  
Available: <https://learn.jquery.com/ajax>.
- [16] «MySQL : MySQL 8.0 Reference Manual : 1.2.1 What is MYSQL?,»  
Available: <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>.
- [17] «What is Entity Framework?,»  
Available: <https://www.entityframeworktutorial.net/what-is-entityframework.aspx>.
- [18] «Get your hands on Entity FrameworkCode First Workflow with new Database and existing Database,»  
Available: <https://www.c-sharpcorner.com/article/get-your-hands-on-entity-framework-code-first-workflow>.