

Report

Οθόνες Εφαρμογής – API - Τεχνολογίες

Λάσκαρης Κωνσταντίνος

itp17405

Τσαγρής Αλέξανδρος

itp17109

Σπυροπούλου Φανή

itp16404



Αθήνα 2019

Εξωτερικό Σύστημα

Η ανάπτυξη του εξωτερικού συστήματος έγινε χρησιμοποιώντας το laravel framework της php, frontend τεχνολογίες javascript και για βάση δεδομένων χρησιμοποιήθηκε η mysql. Για την επικοινωνία με το εσωτερικό σύστημα χρησιμοποιήθηκαν οι resource controller του laravel.

Οθόνες Εφαρμογής

1. /register

Δημιουργήθηκε μοντέλο δεδομένων για τον πολίτη, καθώς και μία φόρμα για να κάνει register στην εξωτερική εφαρμογή. Η παραπάνω διαδικασία γίνεται καλώντας το εσωτερικό σύστημα μέσω API call.

The screenshot shows a web browser window titled 'Election-Mng - Mozilla Firefox' with the address bar showing '127.0.0.1:8000/register'. The page displays a registration form for 'Elections Management'.

Εγγραφή Χρήστη

Όνομα (*) Επίθετο (*) Κωδικός (*)

Email (*) Διεύθυνση (*) Αριθμός (*)

Ταυτότητα / Διαβατήριο (*) # Παραστατικού (*) Παραστατικό (*)

Τράπεζα (*) Iban (*) Δεν θυμάστε το IBAN; Δοκιμάστε αριθμό λογαριασμού

Φύλλο (*) Τηλέφωνο (*) Δημότης (*) Ημ. Γέννησης (*)

Δημιουργία Λογαριασμού

Have an account?

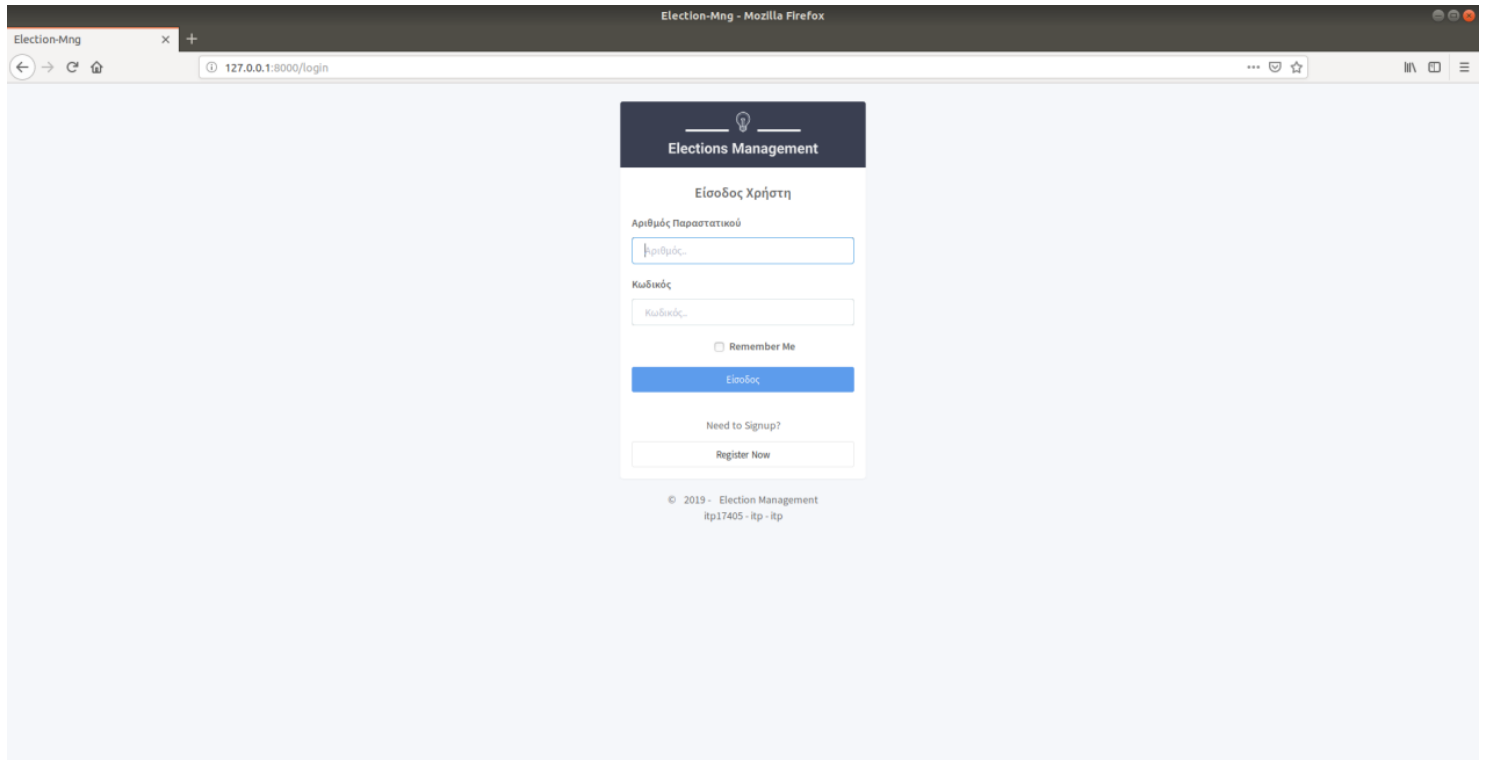
Signup

© 2019 - Election Management
Rpl17405 - Rpl - Rpl

Επίσης, ζητείται ένας λογαριασμός τράπεζας από τον πολίτη. Αν είναι λογαριασμός Εθνικής τράπεζας, τότε μπορεί, πατώντας το κουμπί «Δοκιμάστε αριθμό λογαριασμού» να εισάγει τον αριθμό λογαριασμού και να του εμφανιστεί το iban (μέσω εξωτερικού API call της εθνικής τράπεζας). Επίσης, μόλις πατήσει «Δημιουργία Λογαριασμού», το iban γίνεται validate μέσω εξωτερικού API call της εθνικής τράπεζας. Documentation για τα 2 api: <https://developer.nbg.gr/documentation/IBAN-Validation-API-v12-5476>

2. /login

Ο πολίτης μπορεί να συνδεθεί στην εφαρμογή εισάγοντας τον αριθμός της ταυτότητάς του και τον κωδικό πρόσβασης που δήλωσε στο προηγούμενο βήμα και εφόσον του έχει δοθεί πρόσβαση από τον υπάλληλο του εσωτερικού συστήματος (ελέγχεται μέσω api call), τότε μπορεί να εισέλθει και να δει τις ηλεκτρονικές του υπηρεσίες.



3. /home

Αποτελεί την αρχική σελίδα που βλέπει ο πολίτης. Εκεί εμφανίζεται το ιστορικό του σε αιτήσεις και αλλαγές του προφίλ του. Επίσης, σε περίπτωση που έχει επιλεγεί για εφορευτική επιτροπή, του εμφανίζεται ανάλογο μήνυμα στο επάνω μέρος της οθόνης του. Όλα τα παραπάνω υλοποιούνται με API call στο εσωτερικό σύστημα.

Election-Mng - Mozilla Firefox

127.0.0.1:8000/home

Angle

Main Navigation

- Home
- Το Προφίλ μου
- Αίτηση Ετεροδημότη
- Βεβαίωση Εκλογών

Home Page
Χρήστης: Λάσκαρης Κωστής (id57860)

Ιστορικό Αλλαγών προφίλ

#	Τύπος Ενέργειας	Ημερομηνία Ενέργειας
6	Επεξεργασία Προφίλ	Fri, 24/05/2019 - 16:07:11
7	Επεξεργασία Προφίλ	Fri, 24/05/2019 - 16:07:15

Ιστορικό Αιτήσεων Ετεροδημότη

#	Δημότης	Πόλη	Διεύθυνση	Τηλέφωνο	T.K	Έγγραφο ΔΕΚΟ	Ημερομηνία Ενέργειας	Εκτύπωση
4	ΒΟΙΩΝ/ΚΟΖΑΝΗΣ	Κοζάνης	Κοζανης15	5555555555	12343	storage/eterodimotes/1556703291uid-10.pdf	Fri, 24/05/2019 - 16:08:11	

© 2018 - Angle

4. /profile

Μέσω αυτού του view, ο πολίτης μπορεί να επεξεργαστεί τα στοιχεία του και στη συνέχεια να τα αποθηκεύσει. Τα στοιχεία αποθηκεύονται και στο εξωτερικό, αλλά και στο εσωτερικό σύστημα, μέσω API call.

Election-Mng - Mozilla Firefox

127.0.0.1:8000/profile

Angle

Main Navigation

- Home
- Το Προφίλ μου
- Αίτηση Ετεροδημότη
- Βεβαίωση Εκλογών

Επεξεργασία Προφίλ
Χρήστης: Λάσκαρης Κωστής (id57860)

Επεξεργασία Προφίλ

Όνομα (*) Κωστής

Επίθετο (*) Λάσκαρης

Email (*) k@k.gr

Διεύθυνση (*) Λαρυλαίου

Αριθμός (*) 22

Φύλο (*) Άνδρας

Τηλέφωνο (*) 1111111111

Δημότης (*) ΑΧΑΡΝΩΝ/ΑΤΤΙΚΗΣ

Αποθήκευση Αλλαγών

© 2018 - Angle

5. /eterodimotes

Στο view αυτό, ο πολίτης μπορεί να κάνει αίτηση ετεροδημότη, συμπληρώνοντας τη φόρμα που του παρουσιάζεται. Τα στοιχεία αποθηκεύονται στο εξωτερικό σύστημα και ο πολίτης αφού αποθηκεύσει, μπορεί από την αρχική σελίδα (/home) να εκτυπώσει την αίτησή του.

The screenshot shows a web browser window titled "Election-Mng - Mozilla Firefox" with the address bar displaying "127.0.0.1:8000/eterodimotes". The application interface has a blue header with the "Angle" logo and a user profile "Λάσκαρης Κωστής". A dark sidebar on the left contains navigation links: "Home", "Το Προφίλ μου", "Αίτηση Ετεροδημότη" (highlighted), and "Βεβαίωση Εκλογών". The main content area is titled "Επεξεργασία Προφίλ" and "Λάσκαρης Κωστής (4631860)". It is divided into two panels. The left panel, "Στοιχεία Εκλογέα", contains a form with the following fields: "Αρ. Ταυτότητας: 4657860", "Όνομα: Κωστής", "Επίθετο: Λάσκαρης", "Ημ. Γέννησης: 11/11/1989 (Ετών 30)", "Φύλλο: Άνδρας", and "Δημότης: ΑΧΑΡΝΙΝΩ/ΑΤΤΙΚΗΣ". The right panel, "Καταχώρηση Αίτησης Ετεροδημότη", features a user icon and several required fields marked with an asterisk: "Δημότης (*)" (a dropdown menu), "Πόλη - Περιοχή (*)" (a dropdown menu), "Οδός (*)" (a text field), "Τ.Κ. (*)" (a text field), "Τηλέφωνο (*)" (a text field), "Αριθμός (*)" (a text field), and "Παραστατικό (*)" (a file upload button labeled "Choose file"). A blue button at the bottom of the right panel is labeled "Καταχώρηση Αίτησης". The footer of the application shows "© 2018 - Angle".

6. /voting

Τέλος, ο πολίτης μπορεί, επιλέγοντας από πλαινό μενού της εφαρμογής το «Βεβαίωση Εκλογών», τα κατεβάσει στον Η/Υ μία βεβαίωση προς συμπλήρωση για την άσκηση του εκλογικού του δικαιώματος. Κατά το click, αρχικά ελέγχεται αν ο πολίτης είναι άνω των 18 ετών και στη συνέχεια, αν δεν του έχουν στερηθεί τα εκλογικά του δικαιώματα, μέσω API call στο εσωτερικό σύστημα.

Εσωτερικό Σύστημα

Η ανάπτυξη του εσωτερικού συστήματος έγινε χρησιμοποιώντας το framework του django. Για την βάση δεδομένων χρησιμοποιήθηκε η Sqlite3 python

Υλοποίηση

Αρχικά δημιουργήθηκε το μοντέλο της βάσης (Citizen). Αποθηκεύονται όλα τα στοιχεία των πολιτών με το /register του εξωτερικού συστήματος. Για να αποθηκευτούν τα δεδομένα που στέλνει το εξωτερικό Api δημιουργήθηκε ένα @apiview με post request:

```
@api_view(['POST'])
@permission_classes((IsAuthenticated, ))
def citizen_create(request):
    if request.method == 'POST':
        serializer = CitizenSerializer(data=request.data)
        if serializer.is_valid():
            serializer.save()
            return Response(serializer.data, status=status.HTTP_201_CREATED)
    return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)
```

Μαζί με τα στοιχεία του πολίτη αποθηκεύονται αλλά 3 νέα πεδία:

- 1)Status:Ελέγχεται από τους υπαλλήλους της εφαρμογής η εγκυροτητα των στοιχείων που αποθηκεύονται στην βάση και ανάλογα τον εγκρίνει ή όχι. Αν τον εγκρίνει τότε θα πρέπει να αλλάξει την τιμή του πεδίου από 0(default) σε 1.
- 2)vote_access:Πεδίο που μπορεί να επεξεργαστεί ο υπάλληλος και με την αλλαγή της τιμής του πεδίου μπορεί να στερήσει το εκλογικό δικαίωμα του πολίτη(0 = inactive , 1 = active)
- 3)election: Boolean πεδίο που αφορά την επιλογή του πολίτη για την εφορευτική επιτροπή. Με την αλλαγή της τιμής του πεδίου αυτόματα στέλνεται email που ενημέρωνει τον πολίτη.Δημιουργήθηκε Gmail Λογαριασμος(settings.py).

```
def save(self, *args, **kwargs):
    try:
        if self.number_of_identity:
            old_foo = Citizen.objects.get(pk=self.number_of_identity)
            if old_foo.election == False and self.election == True:
                to_email = [old_foo.email]
                from_email = settings.EMAIL_HOST_USER
                send_mail('ΕΦΟΡΕΥΤΙΚΗ', 'ΕΧΕΤΕ ΕΠΙΛΕΧΤΕΙ ΓΙΑ ΕΦΟΡΕΥΤΙΚΗ ΕΠΙΤΡΟΠΗ', from_email, to_email,
                        fail_silently=False)
    except Citizen.DoesNotExist:
        pass
    super(Citizen, self).save(*args, **kwargs)
```

Για τον έλεγχο των 3 νέων πεδίων απο το εξωτερικό σύστημα δημιουργήθηκε ένα get request:

```
@api_view(['GET'])
@permission_classes((IsAuthenticated, ))
def citizen_list(request, identity):
    if request.method == 'GET':
        citizens_active = Citizen.objects.get(number_of_identity=identity).__dict__
        del citizens_active['_state']
        return JsonResponse(citizens_active, status=status.HTTP_200_OK)
```

Δημιουργήθηκε ακόμα και ένα api με να ενημερώνεται η βάση μας όταν γίνεται επεξεργασία profil ενός πολίτη στο εξωτερικό σύστημα:

```
@api_view(['PUT'])
@permission_classes((IsAuthenticated, ))
def citizen_update(request, identity):
    number_of_identity = identity;
    cit = Citizen.objects.get(pk=number_of_identity)
    serializer = CitizenSerializer(cit, data=request.data)
    if serializer.is_valid():
        serializer.save()
        return Response(serializer.data)
    return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)
```

Τέλος χρησιμοποιήθηκε το Jwt για το authentication των χρηστων . Τα βήματα για την υλοποίηση του είναι τα παρακάτω:

A)δημιουργία ενός super user χρηστη (\$manage.py createsuperuser)

B)generate token (\$manage.py drf_create_token user(εδώ δίνουμε το όνομα που δημιουργήσαμε παραπάνω)

Γ) Σε κάθε request call του εξωτερικού συστήματος θα πρέπει να ακολουθείται και από έναν header:

```
import requests
```

```
url = 'http://127.0.0.1:8001/'nameofapi/'
```

```
headers = {'Authorization': 'Token 'the token''}
```

```
r = requests.get(url, headers=headers)
```

