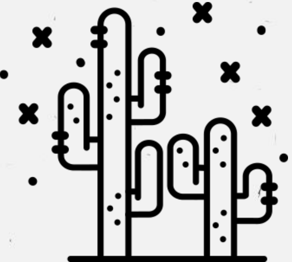
PyWWW

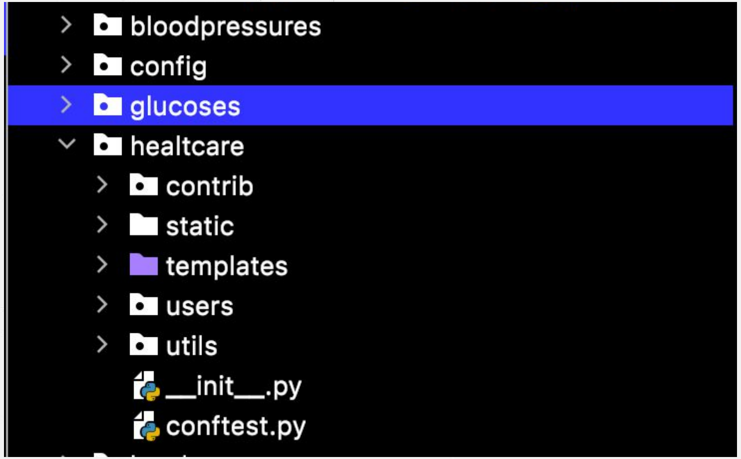
Rozwiązanie pracy domowej



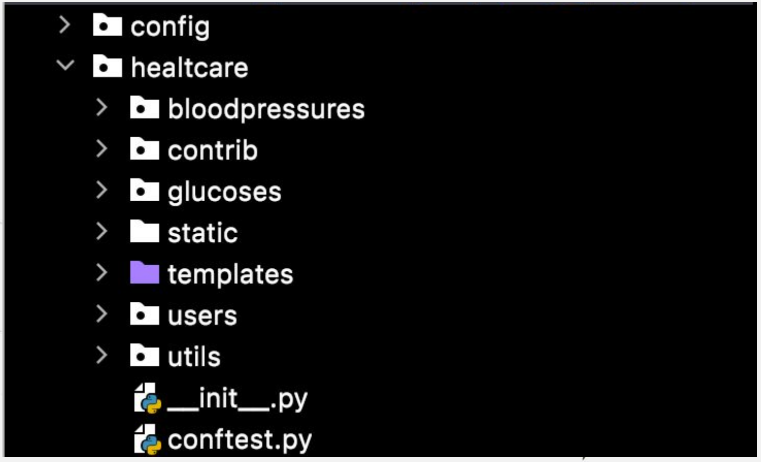
Utworzenie aplikacji

```
$ python manage.py startapp bloodpressures
$ python manage.py startapp glucoses
```

Przeniesienie aplikacji



```
> bloodpressures
> config
> glucoses
v healtcare
    > contrib
    > static
    > templates
    > users
    > utils
      __init__.py
      conftest.py
```



```
> config
v healtcare
    > bloodpressures
    > contrib
    > glucoses
    > static
    > templates
    > users
    > utils
      __init__.py
      conftest.py
```

```
$ mv bloodpressures healthcare
```

Mac / Linux

```
$ move bloodpressures healthcare
```

Windows

Zmiana w apps.py

healthcare/bloodpressures/apps.py

```
from django.apps import AppConfig

class BloodpressuresConfig(AppConfig):

   name = 'healthcare.bloodpressures'
```

healthcare/glucoses/apps.py

```
from django.apps import AppConfig

class GlucosesConfig(AppConfig):

   name = 'healthcare.glucoses'
```

Dodanie do INSTALLED_APPS

config/settings/base.py

```
LOCAL_APPS = [
    "healthcare.users.apps.UsersConfig",
    # Your stuff: custom apps go here
    "healthcare.bloodpressures.apps.BloodpressuresConfig",
    "healthcare.glucoses.apps.GlucosesConfig"
]
```

Dodanie modelu Glucose

healthcare/glucoses/models.py

```
from django.db import models
class Glucose(models.Model):
    user = models.ForeignKey('users.User', on_delete=models.CASCADE)
    value = models.PositiveSmallIntegerField()
    record_datetime = models.DateTimeField()
    notes = models.TextField(null=True, blank=True)
```

Dodanie modelu Bloodpressure

healthcare/bloodpressures/models.py

```python
from django.db import models

class BloodPressure(models.Model):

   user = models.ForeignKey('users.User', on_delete=models.CASCADE)

   systolic = models.PositiveSmallIntegerField()

   diastolic = models.PositiveSmallIntegerField()

   pulse = models.PositiveSmallIntegerField()

   record_datetime = models.DateTimeField()

   notes = models.TextField(null=True, blank=True)
```

Migracje

```
% python manage.py makemigrations
Migrations for 'bloodpressures':
  healthcare/bloodpressures/migrations/0001_initial.py
      - Create model BloodPressure
Migrations for 'glucoses':
  healthcare/glucoses/migrations/0001_initial.py
      - Create model Glucose
$ python manage.py migrate
Operations to perform:
  Apply all migrations: account, admin, auth, bloodpressures, contenttypes, glucoses,
sessions, sites, socialaccount, users
Running migrations:
  Applying bloodpressures.0001_initial... OK
  Applying glucoses.0001_initial... OK
```

Widok listy - bloodpressure

healthcare/bloodpressures/views..py

```python
from django.db.models import QuerySet
from django.views.generic import ListView, DetailView
from django.contrib.auth.mixins import LoginRequiredMixin

from .models import BloodPressure


class BloodPressureListView(LoginRequiredMixin, ListView):

   model = BloodPressure

   def get_queryset(self) -> QuerySet:
       queryset = self.model.objects.filter(user=self.request.user)
       return queryset


bloodpressure_list_view = BloodPressureListView.as_view()
```

Widok szczegółów - bloodpressure

healthcare/bloodpressures/views..py

```
...

class BloodPressureDetailView(LoginRequiredMixin, DetailView):
    model = BloodPressure

bloodpressure_detail_view = BloodPressureDetailView.as_view()
```

urls

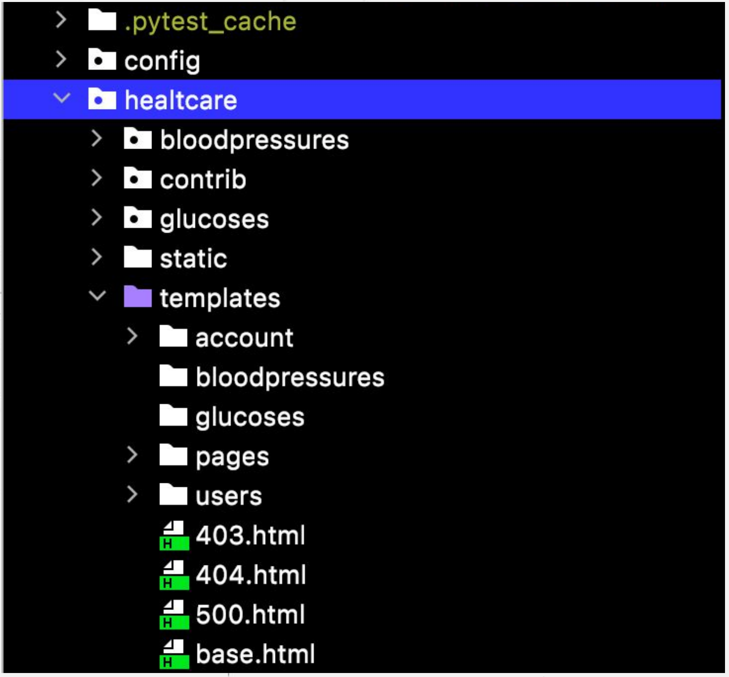
config/urls.py

```
urlpatterns = [
    ...
    # Your stuff: custom urls includes go here
    path("bloodpressures/", include("healthcare.bloodpressures.urls"))
] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

config/urls.py

```
from django.urls import path
from .views import bloodpressure_list_view, bloodpressure_detail_view


app_name = "bloodpressure"
urlpatterns = [
    path("", bloodpressure_list_view, name="list"),
    path("<int:pk>/", bloodpressure_detail_view, name="detail")
]
```

Tworzymy szablony



```
> .pytest_cache
> config
v healtcare
    > bloodpressures
    > contrib
    > glucoses
    > static
    v templates
        > account
          bloodpressures
          glucoses
        > pages
        > users
          403.html
          404.html
          500.html
          base.html
```

PyWWW

Szablon listy - bloodpressure

healthcare/templates/bloodpressures/bloodpressure_list.html

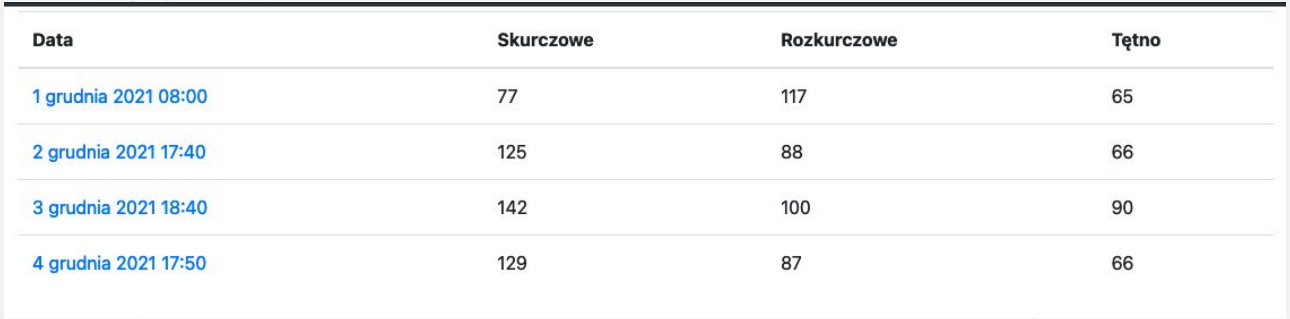
```
{% extends 'base.html' %}

{% block content %}
 <table class="table table-stripped">
   <thead>
   <tr>
     <th scope="col">Data</th>
     <th scope="col">Skurczowe</th>
     <th scope="col">Rozkurczowe</th>
     <th scope="col">Tętno</th>
   </tr>
   {% for bp in object_list %}
     <tr>
       <td><a href="{% url 'bloodpressure:detail' bp.id %}">{{ bp.record_datetime }}</a></td>
       <td>{{ bp.systolic }}</td>
       <td>{{ bp.diastolic }}</td>
       <td>{{ bp.pulse }}</td>
     </tr>
   {% endfor %}
 </thead>
 </table>
{% endblock %}
```

Szablon szczegółów - bloodpressure

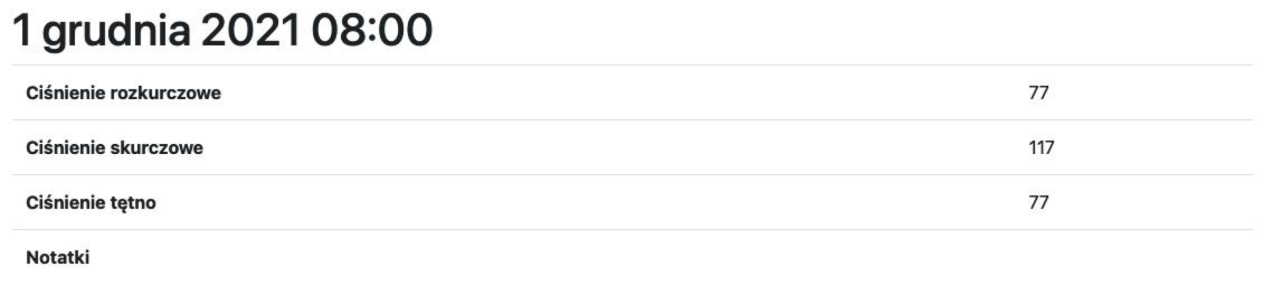
healthcare/templates/bloodpressures/bloodpressure_detail.html

```
{% extends "base.html" %}


{% block content %}
 <h1>{{ object.record_datetime }}</h1>
 <table class="table">
   <tr><th>Ciśnienie rozkurczowe</th><td>{{ object.systolic }}</td></tr>
   <tr><th>Ciśnienie skurczowe</th><td>{{ object.diastolic }}</td></tr>
   <tr><th>Ciśnienie tętno</th><td>{{ object.systolic }}</td></tr>
   <tr><th>Notatki</th><td>{{ object.notes }}</td></tr>
 </table>
{% endblock %}
```

Widok listy i szczegółów w przeglądarce



Data	Skurczowe	Rozkurczowe	Tętno
1 grudnia 2021 08:00	77	117	65
2 grudnia 2021 17:40	125	88	66
3 grudnia 2021 18:40	142	100	90
4 grudnia 2021 17:50	129	87	66



1 grudnia 2021 08:00

Ciśnienie rozkurczowe	77
Ciśnienie skurczowe	117
Ciśnienie tętno	77
Notatki	

Widok listy - glucose

healthcare/glucoses/views..py

```
from django.db.models import QuerySet
from django.views.generic import ListView, DetailView
from django.contrib.auth.mixins import LoginRequiredMixin

from .models import Glucose

class GlucoseListView(LoginRequiredMixin, ListView):

  model = Glucose

  def get_queryset(self) -> QuerySet:

      queryset = self.model.objects.filter(user=self.request.user)

      return queryset


glucose_list_view = GlucoseListView.as_view()
```

Widok szczegółów - glucose

healthcare/bloodpressures/views..py

```
...

class GlucoseDetailView(LoginRequiredMixin, DetailView):
   model = Glucose

glucose_detail_view = GlucoseDetailView.as_view()
```

urls

config/urls.py

```
urlpatterns = [
    ...
    path("glucoses/", include("healthcare.glucoses.urls"))
] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

healthcare/glucoses/urls.py

```
from django.urls import path
from .views import glucose_list_view, glucose_detail_view


app_name = "glucoses"
urlpatterns = [
    path("", glucose_list_view, name="list"),
    path("<int:pk>/", glucose_detail_view, name="detail")
]
```

Szablon listy - glucose

healthcare/templates/glucoses/glucose_list.html

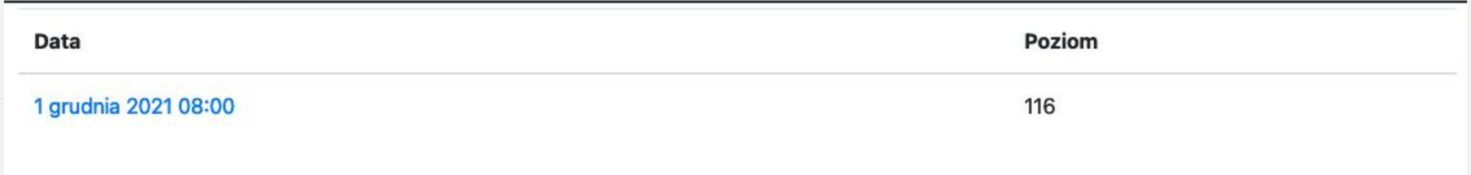
```
{% extends 'base.html' %}

{% block content %}
 <table class="table table-stripped">
  <thead>
  <tr>
   <th scope="col">Data</th>
   <th scope="col">Poziom</th>
  </tr>
  {% for g in object_list %}
   <tr>
    <td><a href="{% url 'glucoses:detail' g.id %}">{{ g.record_datetime }}</a></td>
    <td>{{ g.value }}</td>
   </tr>
  {% endfor %}
 </thead>
 </table>
{% endblock %}
```

Szablon szczegółów - glucose

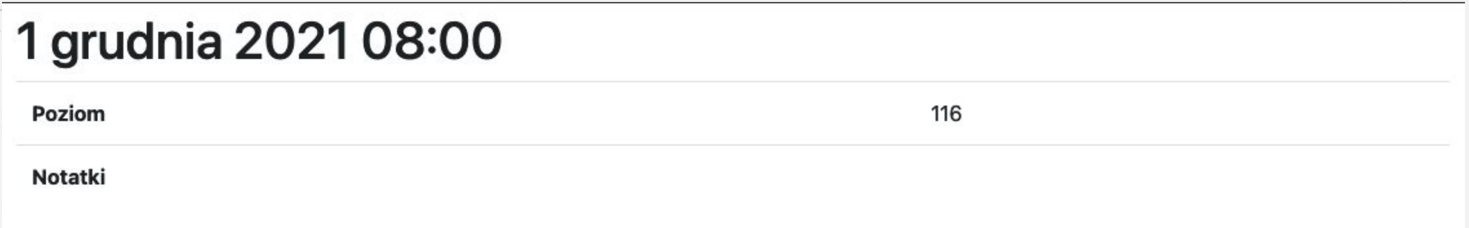
healthcare/templates/glucoses/glucose_detail.html

```
{% extends "base.html" %}


{% block content %}
 <h1>{{ object.record_datetime }}</h1>
 <table class="table">
   <tr><th>Poziom</th><td>{{ object.value }}</td></tr>
   <tr><th>Notatki</th><td>{{ object.notes }}</td></tr>
 </table>
{% endblock %}
```

Widok listy i szczegółów w przeglądarce



Data	Poziom
1 grudnia 2021 08:00	116



1 grudnia 2021 08:00

Poziom	116
Notatki	

Dodawanie - BloodpressueCreateView

healthcare/bloodpressure/views..py

```
from django.urls import reverse_lazy

class BloodPressureCreateView(SuccessMessageMixin, LoginRequiredMixin, CreateView):
   model = BloodPressure
   fields = ['systolic', 'diastolic', 'pulse', 'record_datetime', 'notes']
   success_url = reverse_lazy("bloodpressure:list")
   success_message = "Dodano wpis"

   def get_form(self, form_class=None):
       form = super().get_form(form_class)
       form.fields['record_datetime'].widget = DateTimeInput(
          attrs={'type': 'datetime-local', 'class': 'form-control'}
       )
       return form

   def form_valid(self, form: BaseForm) -> HttpResponse:
       bloodpressure = form.save(commit=False)
       bloodpressure.user = self.request.user
       bloodpressure.save()
       return super().form_valid(form)

bloodpressure_create_view = BloodPressureCreateView.as_view()
```

CreateView - zachowanie domyślne

healthcare/bloodpressure/views..py

```python
class BloodPressureCreateView(CreateView):
   model = BloodPressure
   fields = ['systolic', 'diastolic', 'pulse', 'record_datetime', 'notes']
```

healthcare/bloodpressure/models.py

```python
class BloodPressure(models.Model):
   ...
   def get_absolute_url(self):
       return reverse("bloodpressure:detail", kwargs={"pk": self.id})
```

CreateView - zachowanie domyślne

healthcare/glucoses/models.py

```
from django.db import models

from django.urls import reverse



class Glucose(models.Model):
    ...
    def get_absolute_url(self):
        return reverse("glucoses:detail", args=[self.id])
```

Dodawanie - GlucoseCreateView

healthcare/glucoses/views..py

```python
class GlucoseCreateView(SuccessMessageMixin, LoginRequiredMixin, CreateView):
   model = Glucose
   fields = ['value', 'record_datetime', 'notes']
   success_url = reverse_lazy("glucoses:list")
   success_message = "Dodano pomiar"

   def get_form(self, form_class=None):
       form = super().get_form(form_class)
       form.fields['record_datetime'].widget = DateTimeInput(
           attrs={'type': 'datetime-local', 'class': 'form-control'}
       )
       return form

   def form_valid(self, form: BaseForm) -> HttpResponse:
       glucose = form.save(commit=False)
       glucose.user = self.request.user
       glucose.save()
       return super().form_valid(form)

glucose_create_view = GlucoseCreateView.as_view()
```

urls

healthcare/bloodpressures/urls.py

```
from django.urls import path

from .views import bloodpressure_list_view, bloodpressure_detail_view,
bloodpressure_create_view

app_name = "bloodpressure"
urlpatterns = [
    ...
    path("add/", bloodpressure_create_view, name="add")
]
```

healthcare/glucoses/urls.py

```
from django.urls import path
from .views import glucose_list_view, glucose_detail_view, glucose_create_view

app_name = "glucoses"
urlpatterns = [
    ...
    path("add/", glucose_create_view, name="add"),
]
```

Szablon dodawania - bloodpressure

healthcare/templates/bloodpressures/bloodpressure_form.html

```
{% extends 'base.html' %}

{% load crispy_forms_tags %}

{% block content %}

 <h1>Dodaj pomiar ciśnienia</h1>

 <form method="post" action="{% url 'bloodpressure:add' %}">

   {% csrf_token %}

   {{ form|crispy }}

   <button type="submit" class="btn btn-primary">Dodaj</button>

 </form>

{% endblock %}
```

Szablon dodawania - glucose

healthcare/templates/glucoses/glucose_form.html

```
{% extends 'base.html' %}

{% load crispy_forms_tags %}

{% block content %}

 <h1>Dodaj pomiar cukru</h1>

 <form method="post" action="{% url 'glucoses:add' %}">

   {% csrf_token %}

   {{ form|crispy }}

   <button type="submit" class="btn btn-primary">Dodaj</button>

 </form>

{% endblock %}
```

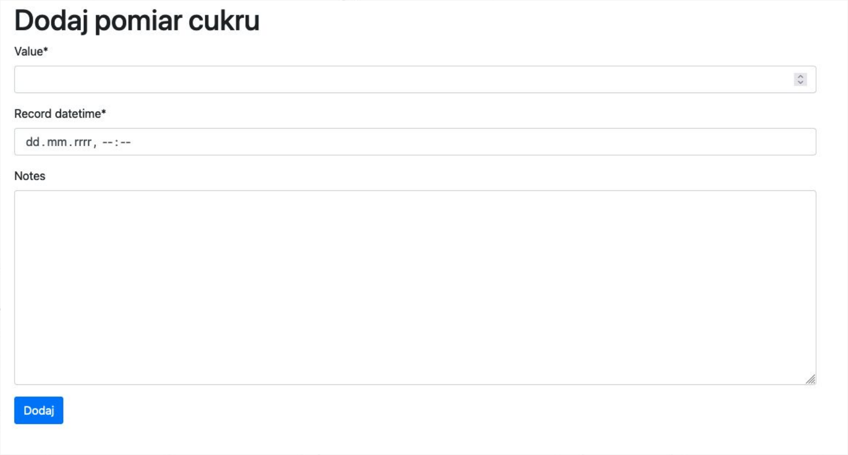
get_absolute_url w szablone

healthcare/templates/glucoses/glucose_list.html

```
<td><a href="{% url 'glucoses:detail' g.id %}">{{ g.record_datetime}}</a></td>
```

```
<td><a href="{{ g.get_absolute_url }}">{{ g.record_datetime }}</a></td>
```

Dodawanie



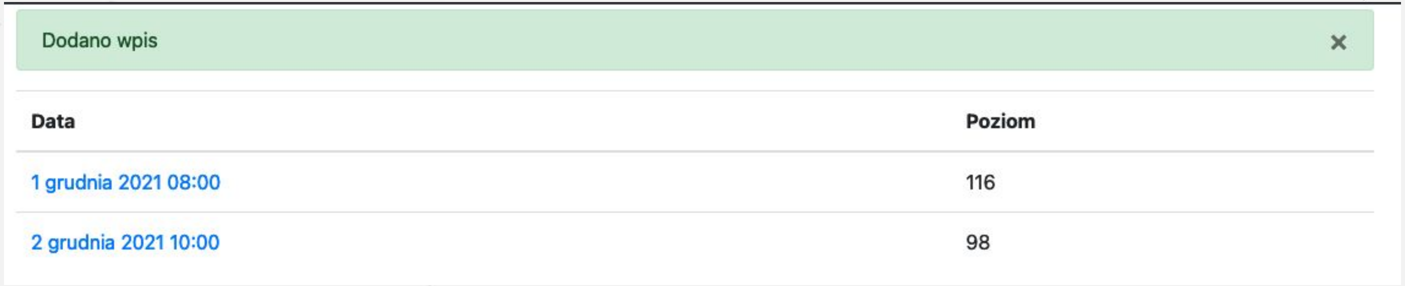
Dodaj pomiar cukru

Value*

Record datetime*

dd.mm.rrrr, --:--

Notes



Dodano wpis

Data	Poziom
1 grudnia 2021 08:00	116
2 grudnia 2021 10:00	98

Dodawanie - BloodpressueUpdateView

healthcare/bloodpressure/views..py

```
class BloodPressureUpdateView(SuccessMessageMixin, LoginRequiredMixin, UpdateView):
   model = BloodPressure
   fields = ['systolic', 'diastolic', 'pulse', 'record_datetime', 'notes']
   success_message = "Zaktualizowano pomiar"
   template_name = 'bloodpressures/bloodpressure_update_form.html'


bloodpressure_update_view = BloodPressureUpdateView.as_view()
```

Szablon aktualizacji - bloodpressure

healthcare/templates/bloodpressures/bloodpressure_update_form.html

```
{% extends 'base.html' %}

{% load crispy_forms_tags %}

{% block content %}

 <h1>Zmień pomiar ciśnienia</h1>

 <form method="post" action="{% url 'bloodpressure:update' %}">

   {% csrf_token %}

   {{ form|crispy }}

   <button type="submit" class="btn btn-primary">Zmień</button>

 </form>

{% endblock %}
```

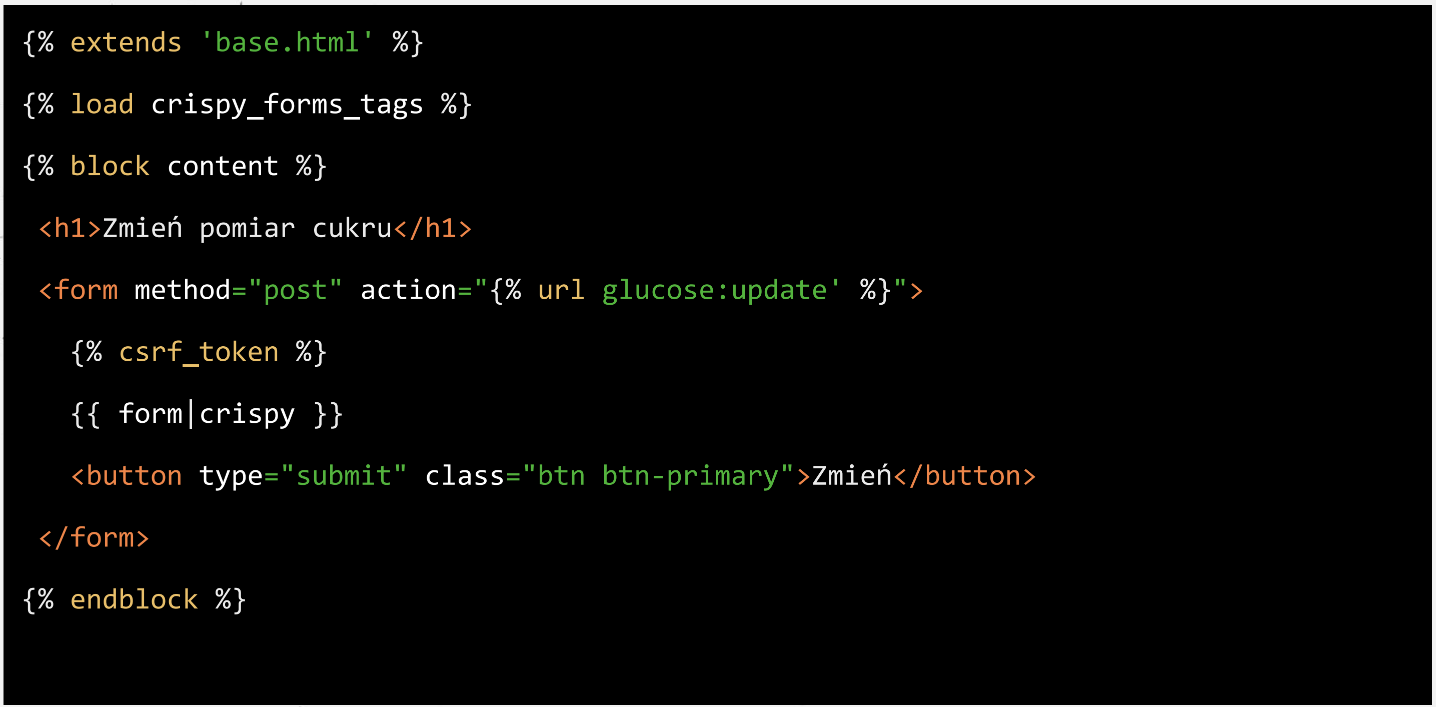
Dodawanie - GlucoseUpdateView

healthcare/glucoses/views..py

```python
class GlucoseUpdateView(SuccessMessageMixin, LoginRequiredMixin, UpdateView):

   model = Glucose

   fields = ['value', 'record_datetime', 'notes']

   success_message = "Zaktualizowano pomiar"

   template_name = 'glucoses/glucose_update_form.html'


glucose_update_view = GlucoseUpdateView.as_view()
```

Szablon aktualizacji - glucose

healthcare/templates/glucoses/glucose_update_form.html



```
{% extends 'base.html' %}

{% load crispy_forms_tags %}

{% block content %}

 <h1>Zmień pomiar cukru</h1>

 <form method="post" action="{% url glucose:update' %}">

   {% csrf_token %}

   {{ form|crispy }}

   <button type="submit" class="btn btn-primary">Zmień</button>

 </form>

{% endblock %}
```

DeleteView - bloodpressure

healthcare/glucoses/views..py

```
class BloodPressureDeleteView(SuccessMessageMixin, LoginRequiredMixin, DeleteView):

    model = BloodPressure

    success_url = reverse_lazy("bloodpressure:list")


bloodpressure_delete_view = BloodPressureDeleteView.as_view()
```

DeleteView - glucose

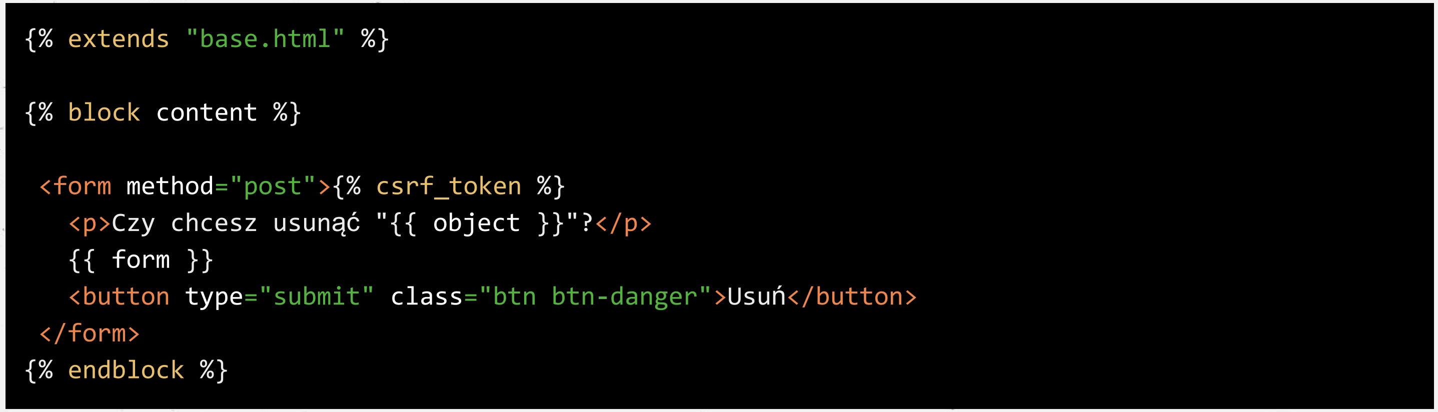
healthcare/glucoses/views..py

```
class GlucoseDeleteView(SuccessMessageMixin, LoginRequiredMixin, DeleteView):
  model = Glucose
  success_url = reverse_lazy("glucoses:list")


glucose_delete_view = GlucoseDeleteView.as_view()
```

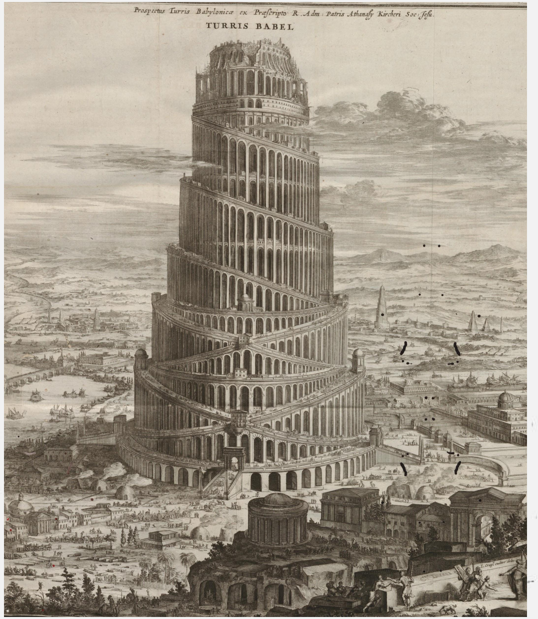
Szablon potwierdzenia

healthcare/templates/bloodpressures/bloodpressure_confirm_delete.html

healthcare/templates/glucoses/glucose_confirm_delete.html



```
{% extends "base.html" %}

{% block content %}

 <form method="post">{% csrf_token %}
   <p>Czy chcesz usunąć "{{ object }}"?</p>
   {{ form }}
   <button type="submit" class="btn btn-danger">Usuń</button>
 </form>
{% endblock %}
```

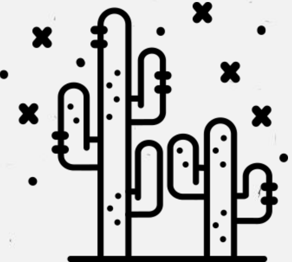

Prospectus Turris Babylonicæ ex Præſcripto R. Adm: Patris Athanaſy Kircheri Soc: Jeſu.
TURRIS BABEL

Tłumaczenia i dokeryzacja

PyWWW

Testowanie aplikacji

Testy



Wykonanie testów

pytest.ini

```
[pytest]
addopts = --ds=config.settings.test --reuse-db -p no:warnings
python_files = tests.py test_*.py
```

https://pytest-django.readthedocs.io/en/latest/database.html

```
$ pytest

Test session starts (platform: darwin, Python 3.7.9, pytest 6.1.0, pytest-sugar 0.9.4)
django: settings: config.settings.test (from option)
rootdir: /Users/pywww/PycharmProjects/django_projects/healthcare, configfile: pytest.ini
plugins: django-test-plus-1.4.0, sugar-0.9.4, django-3.10.0, Faker-10.0.0
collecting ...
 healthcare/users/tests/test_forms.py ✓                                          11% █
 healthcare/users/tests/test_models.py ✓                                         22% ██
 healthcare/users/tests/test_urls.py ✓✓✓                                         56% █████
 healthcare/users/tests/test_views.py ✓✓✓✓                                      100% ██████████

Results (0.77s):
           9 passed
```

permission denied to create database

psql> ALTER ROLE healthcare CREATEDB;
ALTER ROLE

coverage - wykonanie testów

```
 % coverage run -m pytest

Test session starts (platform: darwin, Python 3.7.9, pytest 6.1.0, pytest-sugar 0.9.4)
django: settings: config.settings.test (from option)
rootdir: /Users/pywww/PycharmProjects/django_projects/healthcare, configfile: pytest.ini
plugins: django-test-plus-1.4.0, sugar-0.9.4, django-3.10.0, Faker-10.0.0
collecting ...
 healthcare/users/tests/test_forms.py ✓
      11% ██
 healthcare/users/tests/test_models.py ✓
      22% ███
 healthcare/users/tests/test_urls.py ✓✓✓
      56% ███████
 healthcare/users/tests/test_views.py ✓✓✓✓
      100% █████████████

Results (1.63s):
      9 passed
```

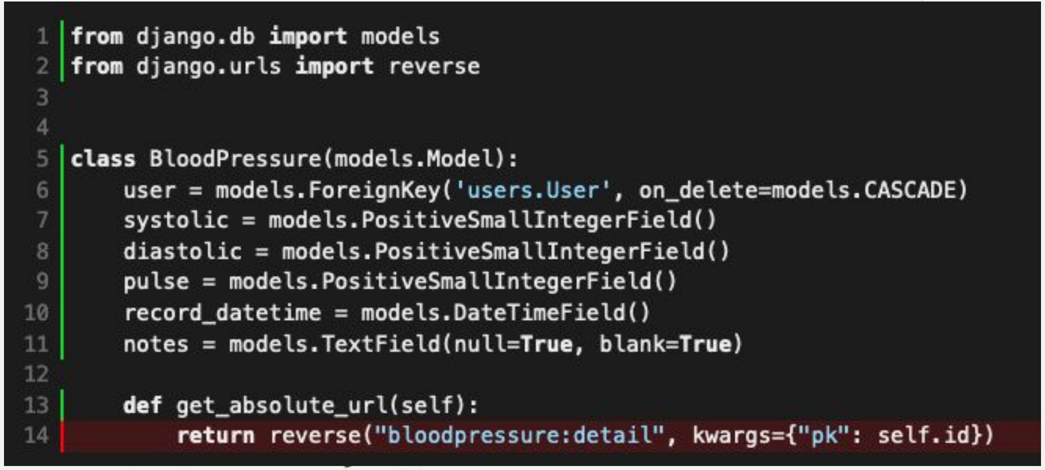
coverage - raport

```
$ coverage report
Name                                 Stmts   Miss  Cover
----------------------------------------------------------
healthcare/__init__.py                      2      0   100%
healthcare/bloodpressures/__init__.py       0      0   100%
healthcare/bloodpressures/admin.py          5      0   100%
healthcare/bloodpressures/apps.py           3      0   100%
healthcare/bloodpressures/models.py         11     1       91%
healthcare/bloodpressures/urls.py           4      0   100%
healthcare/bloodpressures/views.py          41     9       78%
healthcare/conftest.py                      12     0   100%
healthcare/contrib/__init__.py              0      0   100%
healthcare/contrib/sites/__init__.py        0      0   100%
healthcare/glucoses/__init__.py             0      0   100%
healthcare/glucoses/admin.py                5      0   100%
healthcare/glucoses/apps.py                 3      0   100%
healthcare/glucoses/models.py               9      1       89%
healthcare/glucoses/urls.py                 4      0   100%
healthcare/glucoses/views.py                41     9       78%
healthcare/users/__init__.py                0      0   100%
healthcare/users/admin.py                   12     0   100%
healthcare/users/apps.py                    10     0   100%
healthcare/users/forms.py                   18     0   100%
healthcare/users/models.py                  9      0   100%
healthcare/users/urls.py                    4      0   100%
healthcare/users/views.py                   51     10      80%
----------------------------------------------------------
TOTAL                                       244    30      88%
```

coverage - html

```
$ coverage html
$ tree htmlcov
htmlcov
├── coverage_html.js
├── healthcare___init___py.html
├── healthcare_bloodpressures___init___py.html
├── healthcare_bloodpressures_admin_py.html
├── healthcare_bloodpressures_apps_py.html
├── healthcare_bloodpressures_models_py.html
├── healthcare_bloodpressures_urls_py.html
├── healthcare_bloodpressures_views_py.html
├── healthcare_conftest_py.html
├── healthcare_contrib___init___py.html
├── healthcare_contrib_sites___init___py.html
├── healthcare_glucoses___init___py.html
├── healthcare_glucoses_admin_py.html
├── healthcare_glucoses_apps_py.html
├── healthcare_glucoses_models_py.html
├── healthcare_glucoses_urls_py.html
├── healthcare_glucoses_views_py.html
├── healthcare_users___init___py.html
├── healthcare_users_admin_py.html
├── healthcare_users_apps_py.html
├── healthcare_users_forms_py.html
├── healthcare_users_models_py.html
├── healthcare_users_urls_py.html
├── healthcare_users_views_py.html
├── index.html
├── jquery.ba-throttle-debounce.min.js
├── jquery.hotkeys.js
├── jquery.isonscreen.js
├── jquery.min.js
├── jquery.tablesorter.min.js
├── keybd_closed.png
├── keybd_open.png
├── status.json
└── style.css

0 directories, 34 files
```

htmlcov/healthcare_bloodpressures_models_py.html



```python
from django.db import models
from django.urls import reverse


class BloodPressure(models.Model):
    user = models.ForeignKey('users.User', on_delete=models.CASCADE)
    systolic = models.PositiveSmallIntegerField()
    diastolic = models.PositiveSmallIntegerField()
    pulse = models.PositiveSmallIntegerField()
    record_datetime = models.DateTimeField()
    notes = models.TextField(null=True, blank=True)

    def get_absolute_url(self):
        return reverse("bloodpressure:detail", kwargs={"pk": self.id})
```

Lokalizacja testów

```
healthcare/users/tests
├── __init__.py
├── factories.py
├── test_forms.py
├── test_models.py
├── test_urls.py
└── test_views.py
```

testy modelu dla aplikacji users

htmlcov/healthcare_bloodpressures_models_py.html

```python
import pytest

from healthcare.users.models import User

pytestmark = pytest.mark.django_db


def test_user_get_absolute_url(user: User):
    assert user.get_absolute_url() == f"/users/{user.username}/"
```

Fixtury pytest

healthcare/conftest.py

```python
import pytest
from django.test import RequestFactory

from healthcare.users.models import User
from healthcare.users.tests.factories import UserFactory


@pytest.fixture(autouse=True)
def media storage(settings, tmpdir):
    settings.MEDIA_ROOT = tmpdir.strpath


@pytest.fixture
def user() -> User:
    return UserFactory()


@pytest.fixture
def request factory() -> RequestFactory:
    return RequestFactory()
```

Dodanie testu dla model Glucose

1. usunięcie pliku tests.py
2. dodanie package tests w aplikacji
3. utworzenie modułu testów i dopisanie testu

healthcare/glucoses/tests/test_models.py

```
import pytest

from healthcare.glucoses.models import Glucose

pytestmark = pytest.mark.django_db

def test_glucose_get_absolute_url(glucose: Glucose):
    assert glucose.get_absolute_url() == f"/glucoses/{glucose.id}/"
```

Dodanie fixtury dla model Glucose

healthcare/conftest.py

```python
from healthcare.glucoses.tests.factories import GlucoseFactory


@pytest.fixture
def glucose() -> Glucose:
    return GlucoseFactory()
```

Dodanie fabryki dla model Glucose

healthcare/users/tests/factories.py

```python
import random
from datetime import datetime
import factory
from factory.fuzzy import FuzzyDateTime
from pytz import UTC

from healthcare.users.tests.factories import UserFactory


class GlucoseFactory(factory.django.DjangoModelFactory):

    class Meta:
        model = 'glucoses.Glucose'

    user = factory.SubFactory(UserFactory)
    value = factory.LazyAttribute(lambda x: random.randint(60, 350))
    record_datetime = factory.LazyAttribute(lambda x: FuzzyDateTime(
        datetime(2021, 1, 1, tzinfo=UTC)).fuzz()
    )
```

coverage - raport

```
$ coverage report
Name                                 Stmts   Miss  Cover
----------------------------------------------------------
healthcare/__init__.py                       2       0   100%
healthcare/bloodpressures/__init__.py        0       0   100%
healthcare/bloodpressures/admin.py           5       0   100%
healthcare/bloodpressures/apps.py            3       0   100%
healthcare/bloodpressures/models.py          11      1      91%
healthcare/bloodpressures/urls.py            4       0   100%
healthcare/bloodpressures/views.py           41      9      78%
healthcare/conftest.py                       17      0   100%
healthcare/contrib/__init__.py               0       0   100%
healthcare/contrib/sites/__init__.py         0       0   100%
healthcare/glucoses/__init__.py              0       0   100%
healthcare/glucoses/admin.py                 5       0   100%
healthcare/glucoses/apps.py                  3       0   100%
healthcare/glucoses/models.py                9       0   100%
healthcare/glucoses/urls.py                  4       0   100%
healthcare/glucoses/views.py                 41      9      78%
healthcare/users/__init__.py                 0       0   100%
healthcare/users/admin.py                    12      0   100%
healthcare/users/apps.py                     10      0   100%
healthcare/users/forms.py                    18      0   100%
healthcare/users/models.py                   9       0   100%
healthcare/users/urls.py                     4       0   100%
healthcare/users/views.py                    51      10     80%
----------------------------------------------------------
TOTAL                                        249     29     88%
```

PAUZA

ROBIMY

ZADANIE

Zadanie

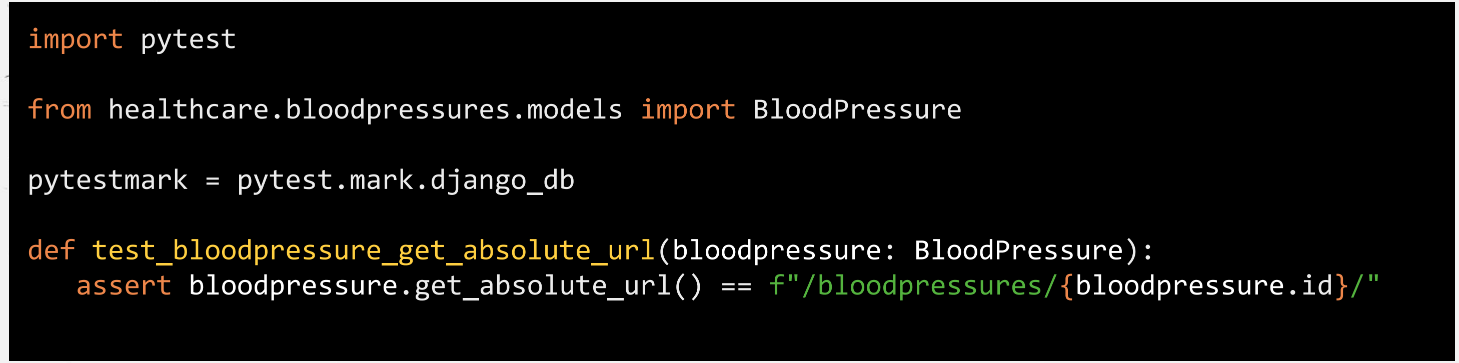
1. Przetestuj get_absolute_url dla bloodpressure

PyWWW

Dodanie testu dla model BloodPressure

1. usunięcie pliku tests.py
2. dodanie package tests w aplikacji
3. utworzenie modułu testów i dopisanie testu

healthcare/bloodpressures/tests/test_models.py



```python
import pytest

from healthcare.bloodpressures.models import BloodPressure

pytestmark = pytest.mark.django_db

def test_bloodpressure_get_absolute_url(bloodpressure: BloodPressure):
   assert bloodpressure.get_absolute_url() == f"/bloodpressures/{bloodpressure.id}/"
```

Dodanie fixtury dla model BloodPressure

healthcare/conftest.py

```python
from healthcare.glucoses.tests.factories import GlucoseFactory


@pytest.fixture
def bloodpressure() -> BloodPressure:
    return BloodPressureFactory()
```

Dodanie fabryki dla model BloodPressure

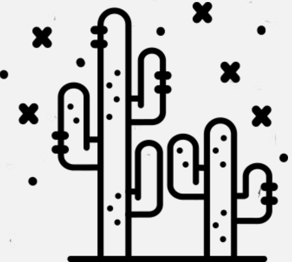
healthcare/users/tests/factories.py

```
from datetime import datetime

import factory
import random

from factory.fuzzy import FuzzyDateTime
from pytz import UTC

from healthcare.users.tests.factories import UserFactory


class BloodPressureFactory(factory.django.DjangoModelFactory):
    class Meta:
        model = 'bloodpressures.BloodPressure'

    user = factory.SubFactory(UserFactory)
    systolic = factory.LazyAttribute(lambda x: random.randint(100, 260))
    # diastolic = factory.LazyAttribute(lambda x: random.randint(40, 120))

    @factory.lazy_attribute
    def diastolic(self):
        diastolic = self.systolic - random.randint(20, 60)
        return diastolic

    pulse = factory.LazyAttribute(lambda x: random.randint(40, 240))
    record_datetime = factory.LazyAttribute(lambda x: FuzzyDateTime(datetime(2021, 1, 1, tzinfo=UTC)).fuzz())
```

PyWWW

Testowanie aplikacji

Testy urls



Testy urls dla User

healthcare/users/tests/test_urls.py

```python
import pytest
from django.urls import reverse, resolve

from healthcare.users.models import User

pytestmark = pytest.mark.django_db


def test_detail(user: User):
   assert reverse("users:detail", kwargs={"username": user.username}) == f"/users/{user.username}/"
   assert resolve(f"/users/{user.username}/").view_name == "users:detail"


def test_update():
   assert reverse("users:update") == "/users/~update/"
   assert resolve("/users/~update/").view_name == "users:update"


def test_redirect():
   assert reverse("users:redirect") == "/users/~redirect/"
   assert resolve("/users/~redirect/").view_name == "users:redirect"
```

Testy urls BloodPressure

healthcare/bloodpressures/tests/test_urls.py

```
...

def test_list():
   assert reverse("bloodpressures:list") == "/bloodpressures/"
   assert resolve("/bloodpressures/").view_name == "bloodpressures:list"


def test_detail(bloodpressure: BloodPressure):
   assert reverse("bloodpressures:detail", kwargs={"pk": bloodpressure.id}) == f"/bloodpressures/{bloodpressure.id}/"
   assert resolve(f"/bloodpressures/{bloodpressure.id}/").view_name == "bloodpressures:detail"


def test_create():
   assert reverse("bloodpressures:add") == "/bloodpressures/add/"
   assert resolve("/bloodpressures/add/").view_name == "bloodpressures:add"


def test_update(bloodpressure: BloodPressure):
   assert reverse("bloodpressures:update", args=[bloodpressure.id]) == f"/bloodpressures/{bloodpressure.id}/update/"
   assert resolve(f"/bloodpressures/{bloodpressure.id}/update/").view_name == "bloodpressures:update"


def test_delete(bloodpressure: BloodPressure):
   assert reverse("bloodpressures:delete", args=[bloodpressure.id]) == f"/bloodpressures/{bloodpressure.id}/dele
   assert resolve(f"/bloodpressures/{bloodpressure.id}/delete/").view_name == "bloodpressures:delete"
```

Testu urls BloodPressure

```
$ pytest
```

```
Results (1.49s):
       11 passed
       5 failed
       - healthcare/glucoses/tests/test urls.py:9 test list
       - healthcare/glucoses/tests/test_urls.py:14 test_detail
       - healthcare/glucoses/tests/test urls.py:19 test create
       - healthcare/glucoses/tests/test_urls.py:24 test_update
       - healthcare/glucoses/tests/test_urls.py:29 test_delete
```

django.urls.exceptions.NoReverseMatch: 'bloodpressures' is not a registered namespace

Testu urls BloodPressure

healthcare/bloodpressures/urls.py

```
app_name = "bloodpressure"
urlpatterns = [
    ...
]
```

```
app_name = "bloodpressures"
urlpatterns = [
    ...
]
```

Testu urls BloodPressure

```
$ pytest
```

```
Results (0.88s):
       14 passed
       2 failed
       - healthcare/bloodpressures/tests/test_urls.py:24 test_update
       - healthcare/bloodpressures/tests/test_urls.py:29 test_delete
```

```
       def test_update(bloodpressure: BloodPressure):
>      assert reverse("bloodpressures:update", args=[bloodpressure.id]) ==
f"/bloodpressures/{bloodpressure.id}/update/"
E      AssertionError: assert '/bloodpressures/38/update' == '/bloodpressures/38/update/'
E      - /bloodpressures/38/update/
E      ?                          -
E      + /bloodpressures/38/update
```

Poprawki urls BloodPressure

healthcare/bloodpressures/urls.py

```
app_name = "bloodpressures"
urlpatterns = [
    ...
    path("<int:pk>/update", bloodpressure_update_view, name="update"),
    path("<int:pk>/delete", bloodpressure_delete_view, name="delete"),
]
```

```
app_name = "bloodpressures"
urlpatterns = [
    ...
    path("<int:pk>/update/", bloodpressure_update_view, name="update"),
    path("<int:pk>/delete/", bloodpressure_delete_view, name="delete"),
]
```

PAUZA

ROBIMY

ZADANIE

Zadanie

1. Zrób testy urls dla aplikacji glucoses

Testy urls BloodPressure

healthcare/glucoses/tests/test_urls.py

```
...

def test_list():
   assert reverse("glucoses:list") == "/glucoses/"
   assert resolve("/glucoses/").view_name == "glucoses:list"


def test_detail(glucose: Glucose):
   assert reverse("glucoses:detail", kwargs={"pk": glucose.id}) == f"/glucoses/{glucose.id}/"
   assert resolve(f"/glucoses/{glucose.id}/").view_name == "glucoses:detail"


def test_create():
   assert reverse("glucoses:add") == "/glucoses/add/"
   assert resolve("/glucoses/add/").view_name == "glucoses:add"


def test_update(glucose: Glucose):
   assert reverse("glucoses:update", args=[glucose.id]) == f"/glucoses/{glucose.id}/update/"
   assert resolve(f"/glucoses/{glucose.id}/update/").view_name == "glucoses:update"


def test_delete(glucose: Glucose):
   assert reverse("glucoses:delete", args=[glucose.id]) == f"/glucoses/{glucose.id}/delete/"
   assert resolve(f"/glucoses/{glucose.id}/delete/").view_name == "glucoses:delete"
```

Poprawki urls Glucose

healthcare/bloodpressures/urls.py

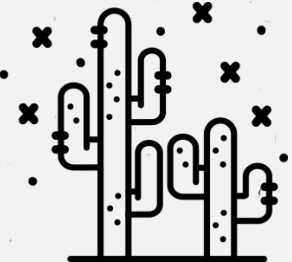
```
app_name = "glucoses"
urlpatterns = [
    ...
    path("<int:pk>/update", glucose_update_view, name="update"),
    path("<int:pk>/delete", glucose_delete_view, name="delete"),
]
```

```
app_name = "glucoses"
urlpatterns = [
    ...
    path("<int:pk>/update/", glucose_update_view, name="update"),
    path("<int:pk>/delete/", glucose_delete_view, name="delete"),
]
```

PyWWW

Testowanie aplikacji

Testy widoków



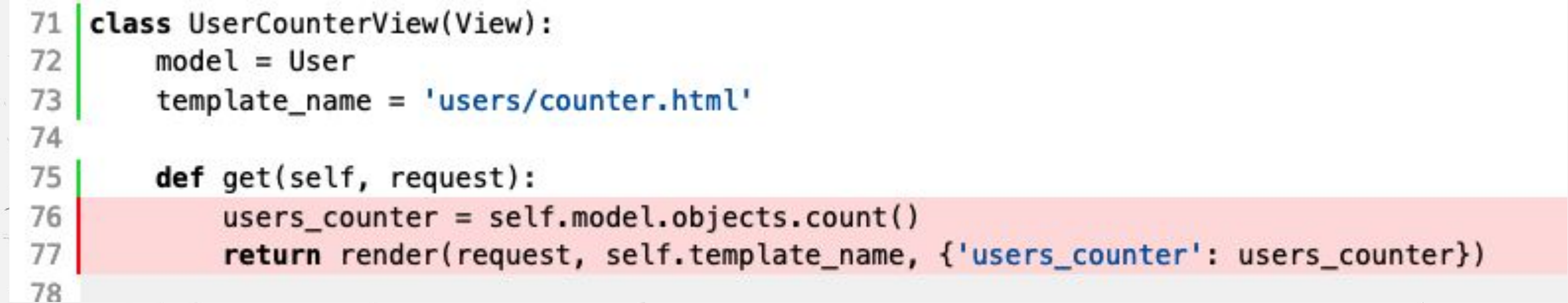
Testy widoków w Users

healthcare/users/tests/test_views.py

```
class TestUserUpdateView:

    def test_get_success_url(
        self, user: User, request_factory: RequestFactory
    ):
        ...

    def test_get_object(
        self, user: User, request_factory: RequestFactory
    ):
        ...

    def test_form_valid(
        self, user: User, request_factory: RequestFactory
    ):
        ...
```

Testy widoków w Users

healthcare/htmlcov/healthcare_users_views_py.html



```
class UserCounterView(View):
    model = User
    template_name = 'users/counter.html'

    def get(self, request):
        users_counter = self.model.objects.count()
        return render(request, self.template_name, {'users_counter': users_counter})
```

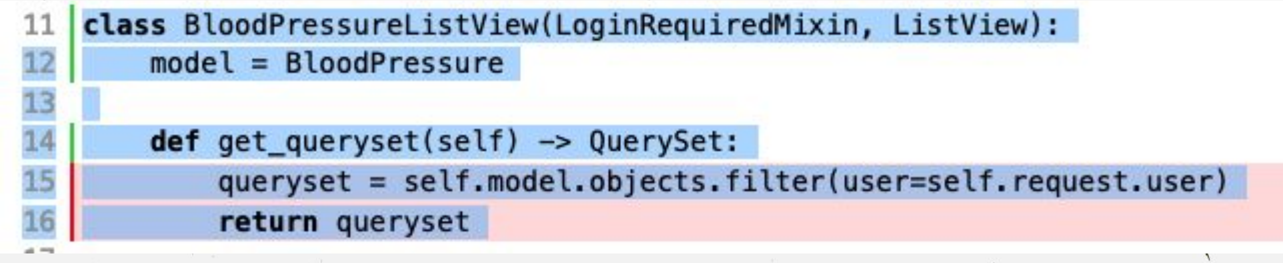
Testy widoków w Users

healthcare/users/tests/test_views.py

```python
from django.test import client

class TestUserCounterView:

   def test_context_without_one_user(self, user):
       c = client.Client()
       response = c.get(reverse('users:counter'))

       assert 'users_counter' in response.context
       assert response.context.get('users_counter') == 1

   def test_context_with_2_users(self, user: User):

       UserFactory()

       c = client.Client()
       response = c.get(reverse('users:counter'))

       assert 'users_counter' in response.context
       assert response.context.get('users_counter') == 2
```

Testy widoków w bloodpressures

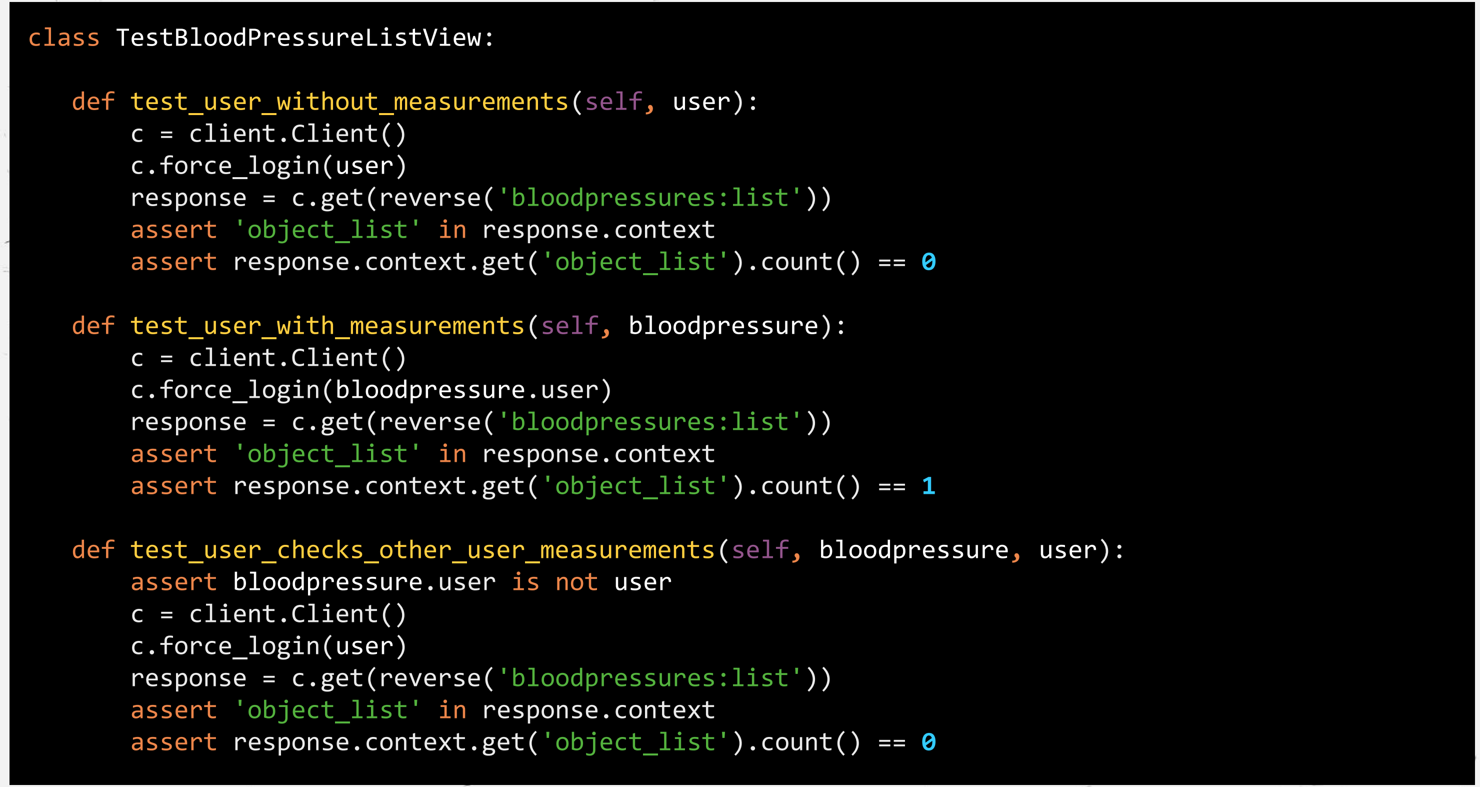
healthcare/htmlcov/healthcare_bloodpressures_views_py.html



```python
class BloodPressureListView(LoginRequiredMixin, ListView):
    model = BloodPressure

    def get_queryset(self) -> QuerySet:
        queryset = self.model.objects.filter(user=self.request.user)
        return queryset
```

Testy widoków w BloodPressure

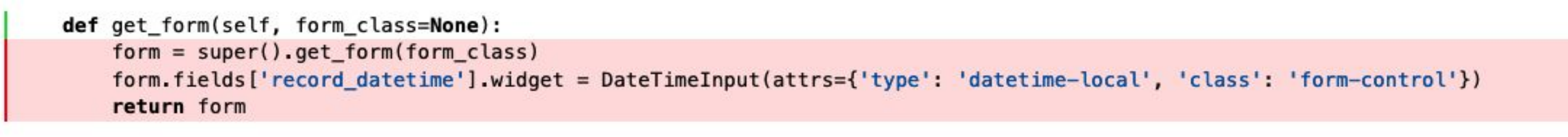
healthcare/bloodpressures/tests/test_views.py



```python
class TestBloodPressureListView:

    def test_user_without_measurements(self, user):
        c = client.Client()
        c.force_login(user)
        response = c.get(reverse('bloodpressures:list'))
        assert 'object_list' in response.context
        assert response.context.get('object_list').count() == 0

    def test_user_with_measurements(self, bloodpressure):
        c = client.Client()
        c.force_login(bloodpressure.user)
        response = c.get(reverse('bloodpressures:list'))
        assert 'object_list' in response.context
        assert response.context.get('object_list').count() == 1

    def test_user_checks_other_user_measurements(self, bloodpressure, user):
        assert bloodpressure.user is not user
        c = client.Client()
        c.force_login(user)
        response = c.get(reverse('bloodpressures:list'))
        assert 'object_list' in response.context
        assert response.context.get('object_list').count() == 0
```

Testy widoków w bloodpressures

healthcare/htmlcov/healthcare_bloodpressures_views_py.html



```
    def get_form(self, form_class=None):
        form = super().get_form(form_class)
        form.fields['record_datetime'].widget = DateTimeInput(attrs={'type': 'datetime-local', 'class': 'form-control'})
        return form
```

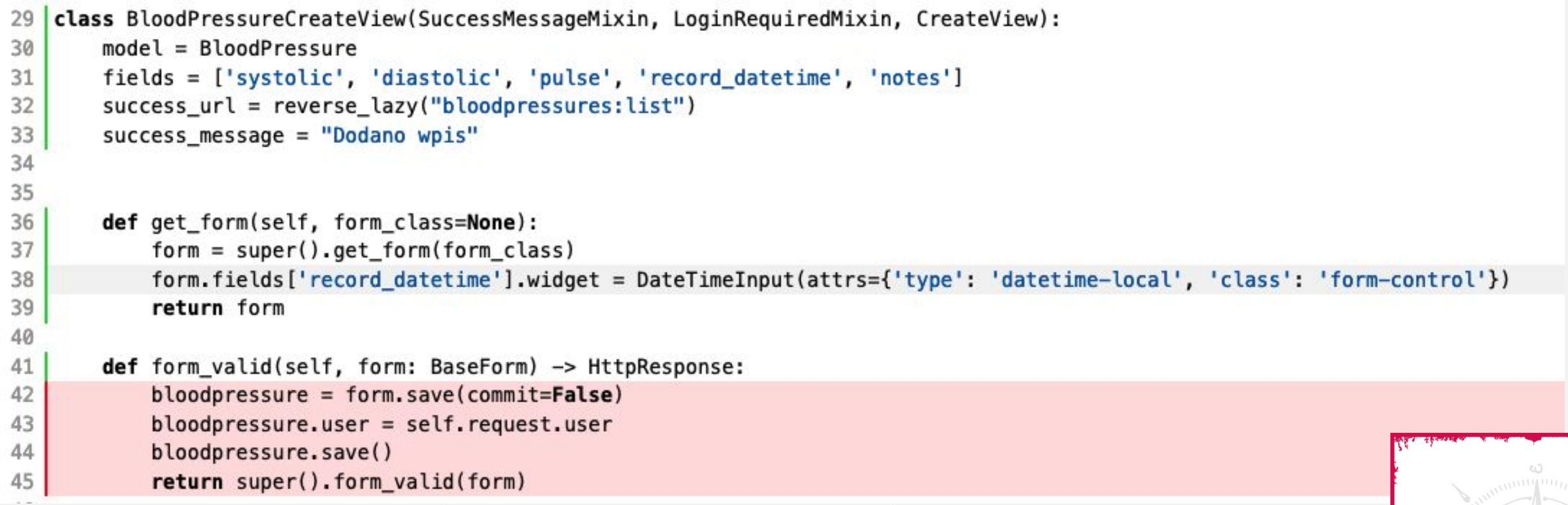
Testy widoków w BloodPressure

healthcare/bloodpressures/tests/test_views.py

```
class TestBloodPressureCreateView:

    def test_get_form(self, request_factory: RequestFactory):
        view = BloodPressureCreateView()
        request = request_factory.get(reverse("bloodpressures:add"))
        view.setup(request)
        form = view.get_form()
        assert isinstance(form.fields['record_datetime'].widget, DateTimeInput)
```

Testy widoków w bloodpressures

healthcare/htmlcov/healthcare_bloodpressures_views_py.html



```python
class BloodPressureCreateView(SuccessMessageMixin, LoginRequiredMixin, CreateView):
    model = BloodPressure
    fields = ['systolic', 'diastolic', 'pulse', 'record_datetime', 'notes']
    success_url = reverse_lazy("bloodpressures:list")
    success_message = "Dodano wpis"


    def get_form(self, form_class=None):
        form = super().get_form(form_class)
        form.fields['record_datetime'].widget = DateTimeInput(attrs={'type': 'datetime-local', 'class': 'form-control'})
        return form

    def form_valid(self, form: BaseForm) -> HttpResponse:
        bloodpressure = form.save(commit=False)
        bloodpressure.user = self.request.user
        bloodpressure.save()
        return super().form_valid(form)
```

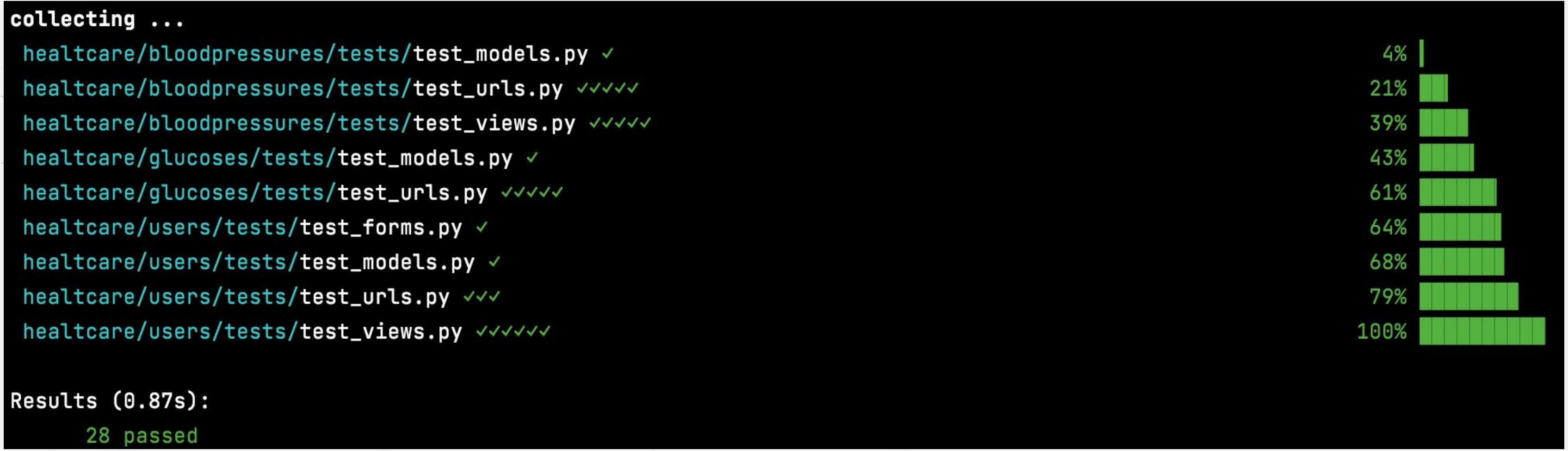
PyWWW

Testy widoków w BloodPressure

healthcare/bloodpressures/tests/test_views.py

```
...
from django.contrib.messages.middleware import MessageMiddleware
from django.contrib.sessions.middleware import SessionMiddleware
from django.utils import timezone
...

def test_form_valid(self, user, request_factory):
   data = {'systolic': 140, 'diastolic': 105, 'pulse': 70, 'record_datetime': timezone.now()}
   request = request_factory.post(reverse("bloodpressures:add"), data=data)
   request.user = user
   session_middleware = SessionMiddleware()
   session_middleware.process_request(request)
   msg_middleware = MessageMiddleware()
   msg_middleware.process_request(request)
   response = BloodPressureCreateView.as_view()(request)
   user.refresh_from_db()

   assert response.status_code == 302
   bp = user.bloodpressure_set.last()
   assert bp.systolic == data.get('systolic')
   assert bp.diastolic == data.get('diastolic')
   assert bp.pulse == data.get('pulse')
   assert bp.record_datetime == data.get('record_datetime')
```

Stan testów i pokrycia



```
collecting ...
 healtcare/bloodpressures/tests/test_models.py ✓                                                                           4%
 healtcare/bloodpressures/tests/test_urls.py ✓✓✓✓✓                                                                        21%
 healtcare/bloodpressures/tests/test_views.py ✓✓✓✓✓                                                                       39%
 healtcare/glucoses/tests/test_models.py ✓                                                                                43%
 healtcare/glucoses/tests/test_urls.py ✓✓✓✓✓                                                                              61%
 healtcare/users/tests/test_forms.py ✓                                                                                    64%
 healtcare/users/tests/test_models.py ✓                                                                                   68%
 healtcare/users/tests/test_urls.py ✓✓✓                                                                                   79%
 healtcare/users/tests/test_views.py ✓✓✓✓✓✓                                                                              100%

Results (0.87s):
      28 passed
```

Stan testów i pokrycia

```
% coverage report
Name                                    Stmts   Miss  Cover
-----------------------------------------------------------
healthcare/__init__.py                         2     0   100%
healthcare/bloodpressures/__init__.py          0     0   100%
healthcare/bloodpressures/admin.py             5     0   100%
healthcare/bloodpressures/apps.py              3     0   100%
healthcare/bloodpressures/models.py                  11     0   100%
healthcare/bloodpressures/urls.py              4     0   100%
healthcare/bloodpressures/views.py                   41     0   100%
healthcare/conftest.py                               22     0   100%
healthcare/contrib/__init__.py                 0     0   100%
healthcare/contrib/sites/__init__.py           0     0   100%
healthcare/glucoses/__init__.py                0     0   100%
healthcare/glucoses/admin.py                   5     0   100%
healthcare/glucoses/apps.py                    3     0   100%
healthcare/glucoses/models.py                  9     0   100%
healthcare/glucoses/urls.py                    4     0   100%
healthcare/glucoses/views.py                         41     9      78%
healthcare/users/__init__.py                   0     0   100%
healthcare/users/admin.py                            12     0   100%
healthcare/users/apps.py                             10     0   100%
healthcare/users/forms.py                      18    0   100%
healthcare/users/models.py                     9     0   100%
healthcare/users/urls.py                       4     0   100%
healthcare/users/views.py                      42    0   100%
healthcare/utils/__init__.py                   0     0   100%
healthcare/utils/context_processors.py         3     0   100%
-----------------------------------------------------------
TOTAL                                   248    9     96%
```

PAUZA

ROBIMY

ZADANIE

Zadanie

1. Napisz testy widków dla glucoses

PyWWW

PyWWW

Testowanie aplikacji

Testy funkcjonalne

Selenium

https://www.selenium.dev/

Selenium z pythonem:

https://selenium-python.readthedocs.io/

Web driver:

https://github.com/mozilla/geckodriver

```
$ pip install selenium
```

Przykładowy test z Selenium

healthcare/bloodpressures/tests/test_views.py

```python
from django.test import LiveServerTestCase
from selenium import webdriver
from selenium.webdriver.common.by import By


class TestForNotLoggedInUser(LiveServerTestCase):

    def test_main_page(self):
        driver = webdriver.Firefox()
        driver.get(self.live_server_url)
        element = driver.find_element(By.ID, 'log-in-link')
        assert "Zaloguj" in element.text
        driver.close()
```

selenium.webdriver.common.by

```python
class By(object):
    """
    Set of supported locator strategies.
    """

    ID = "id"
    XPATH = "xpath"
    LINK_TEXT = "link text"
    PARTIAL_LINK_TEXT = "partial link text"
    NAME = "name"
    TAG_NAME = "tag name"
    CLASS_NAME = "class name"
    CSS_SELECTOR = "css selector"
```

https://docs.djangoproject.com/en/4.0/topics/testing/tools/#django.test.LiveServerTestCase

Przykładowy test z Selenium

healthcare/bloodpressures/tests/test_functionals.py

```python
from django.test import LiveServerTestCase
from selenium import webdriver
from selenium.webdriver.common.by import By


class TestForNotLoggedInUser(LiveServerTestCase):

   def test_main_page(self):
       driver = webdriver.Firefox()
       driver.get(self.live_server_url)
       element = driver.find_element(By.ID, 'log-in-link')
       assert "Zaloguj" in element.text
       driver.close()
```

selenium.webdriver.common.by

```python
class By(object):
    """
    Set of supported locator strategies.
    """

    ID = "id"
    XPATH = "xpath"
    LINK_TEXT = "link text"
    PARTIAL_LINK_TEXT = "partial link text"
    NAME = "name"
    TAG_NAME = "tag name"
    CLASS_NAME = "class name"
    CSS_SELECTOR = "css selector"
```

https://docs.djangoproject.com/en/4.0/topics/testing/tools/#django.test.LiveServerTestCase

PyWWW

Przykładowy test z Selenium

```
pip install django-selenium-login
```

healthcare/bloodpressures/tests/test_functionals.py

```
from django.test import LiveServerTestCase
from selenium import webdriver
from selenium.webdriver.common.by import By
from seleniumlogin import force_login

from healthcare.users.tests.factories import UserFactory


class TestForLoggedInUser(LiveServerTestCase):

   def test_main_page(self):
       driver = webdriver.Firefox()
       user = UserFactory()
       force_login(user, driver, self.live_server_url)
       driver.get(self.live_server_url)
       element = driver.find_element(By.CSS_SELECTOR, 'ul li.nav-item:last-child a')

       assert "Wyloguj" in element.text
       driver.quit()
```

PAUZA

ROBIMY

ZADANIE

Zadanie

Napisz test, który pokaże, że jeśli niezalogowany użytkownik spróbuje wejść na listę pomiarów glukozy, to zostanie przekierowany na stronę logowania.

Rozwiązanie

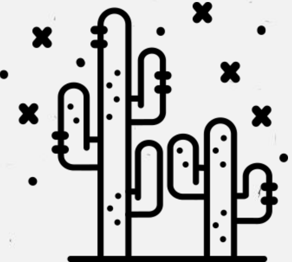
healthcare/bloodpressures/tests/test_functionals.py

```
...
from django.urls import reverse
...

class TestForNotLoggedInUser(LiveServerTestCase):

    def test_main_page(self):
        …

    def test_glucoses_list_page(self):
        driver = webdriver.Firefox()
        driver.get(self.live_server_url + reverse('glucoses:list'))
        expected_url = self.live_server_url + '/accounts/login/?next=/glucoses/'
        assert driver.current_url == expected_url
        driver.quit()
```

PyWWW

Puszczamy aplikację w świat!

Deployment



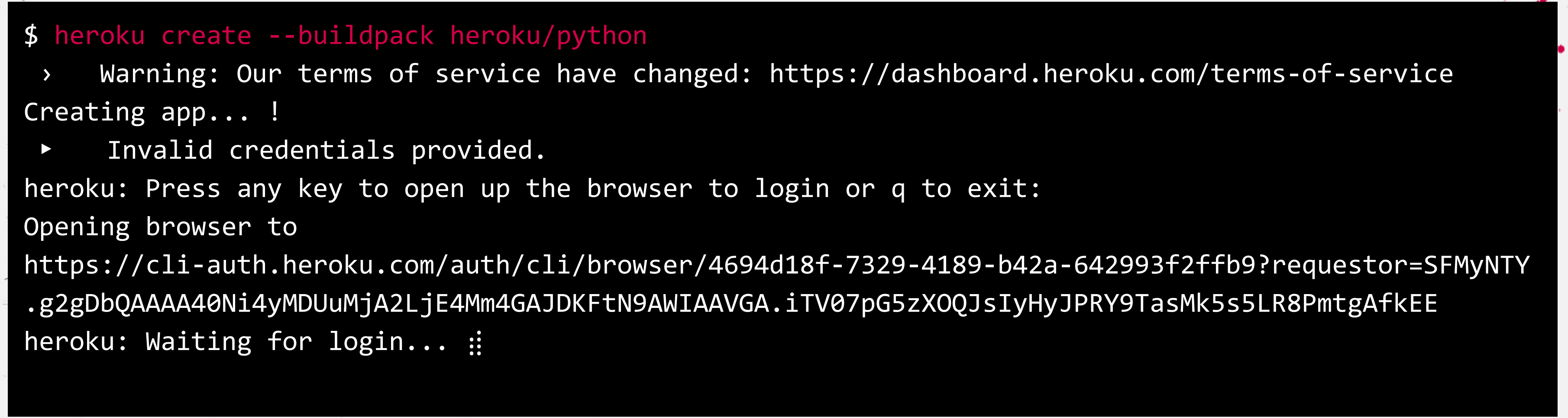
Instalacja klienta heroku

HEROKU

https://devcenter.heroku.com/articles/heroku-cli#download-and-install

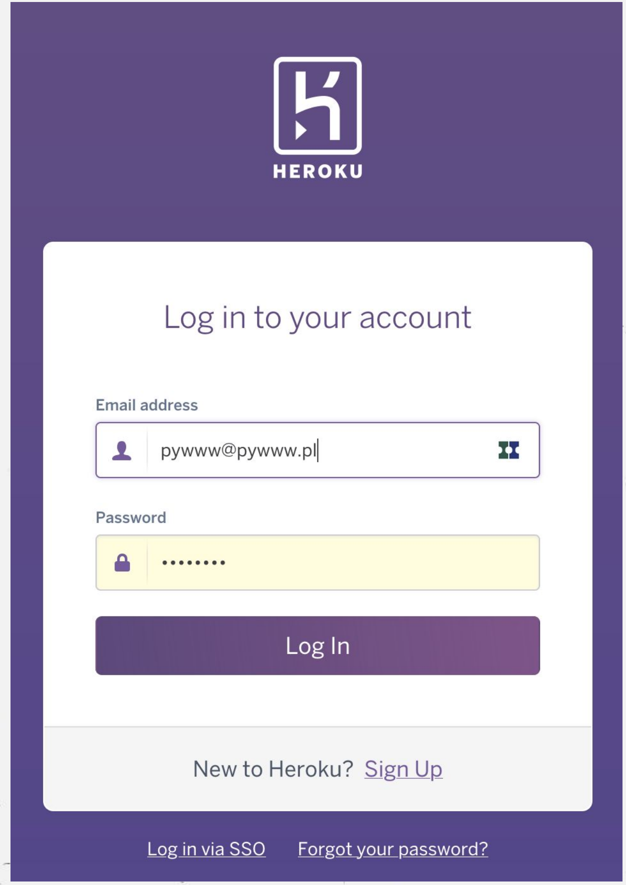
PyWWW

Instalacja klienta heroku



```
$ heroku create --buildpack heroku/python
 ›   Warning: Our terms of service have changed: https://dashboard.heroku.com/terms-of-service
Creating app... !
 ▸    Invalid credentials provided.
heroku: Press any key to open up the browser to login or q to exit:
Opening browser to
https://cli-auth.heroku.com/auth/cli/browser/4694d18f-7329-4189-b42a-642993f2ffb9?requestor=SFMyNTY
.g2gDbQAAAA40Ni4yMDUuMjA2LjE4Mm4GAJDKFtN9AWIAAVGA.iTV07pG5zXOQJsIyHyJPRY9TasMk5s5LR8PmtgAfkEE
heroku: Waiting for login... ⣾
```

Instalacja klienta heroku



HEROKU

Log in to your account

Email address

pywww@pywww.pl

Password

•••••••••

New to Heroku? Sign Up

Log in via SSO   Forgot your password?

Inicjalizacja projektu

```
Creating app... done, ⬢ rocky-anchorage-37573
Setting buildpack to heroku/python... done
https://rocky-anchorage-37573.herokuapp.com/ |
https://git.heroku.com/rocky-anchorage-37573.git
```

Dodanie dodatku do postgres

```
$ heroku addons:create heroku-postgresql:hobby-dev

Creating heroku-postgresql:hobby-dev on ⬢ rocky-anchorage-37573... free
Database has been created and is available
 ! This database is empty. If upgrading, you can transfer
 ! data from another database with pg:copy
Created postgresql-rectangular-31559 as DATABASE_URL
Use heroku addons:docs heroku-postgresql to view documentation
```

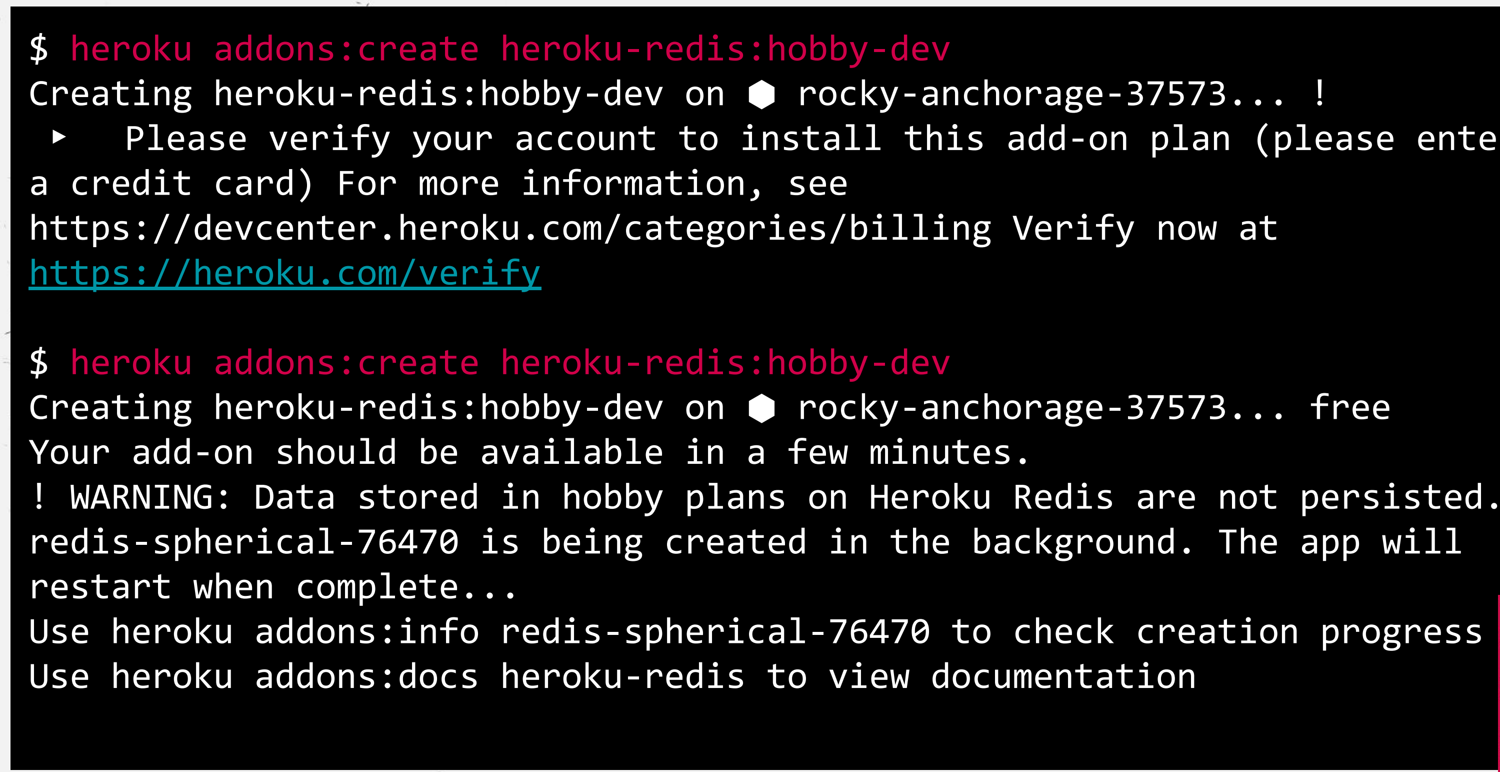
Ustawienie backupy

```
$ heroku pg:backups schedule --at '02:00 America/Los_Angeles' DATABASE_URL

Scheduling automatic daily backups of postgresql-rectangular-31559 at 02:00
America/Los_Angeles... done
```

Ustawienie postgres

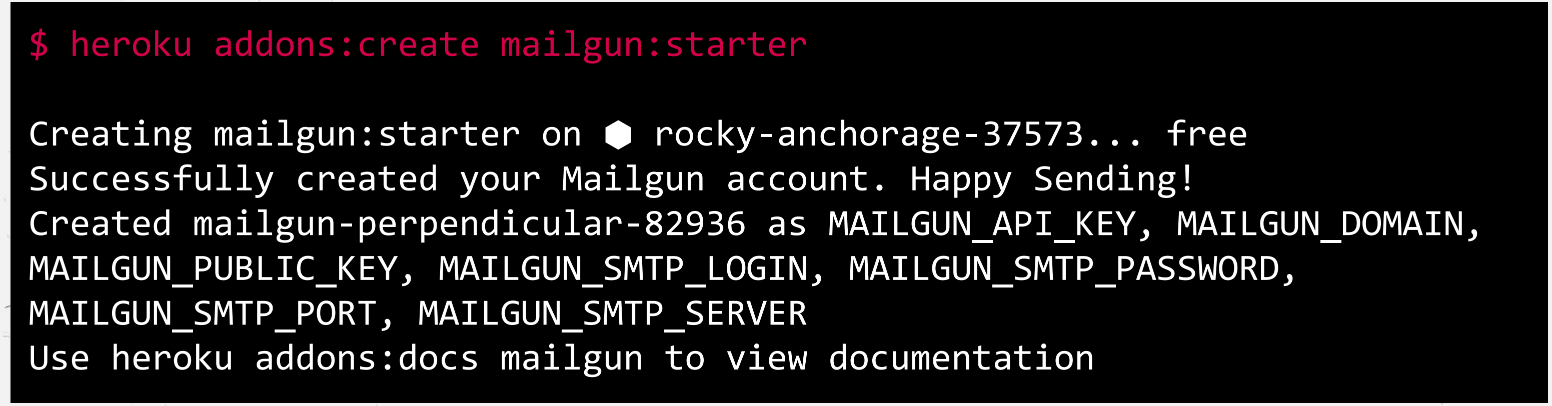
```
$ heroku pg:promote DATABASE_URL

Ensuring an alternate alias for existing DATABASE_URL... !
 ▸   postgresql-rectangular-31559 is already promoted on ⬢
rocky-anchorage-37573
```

Ustawienie redis



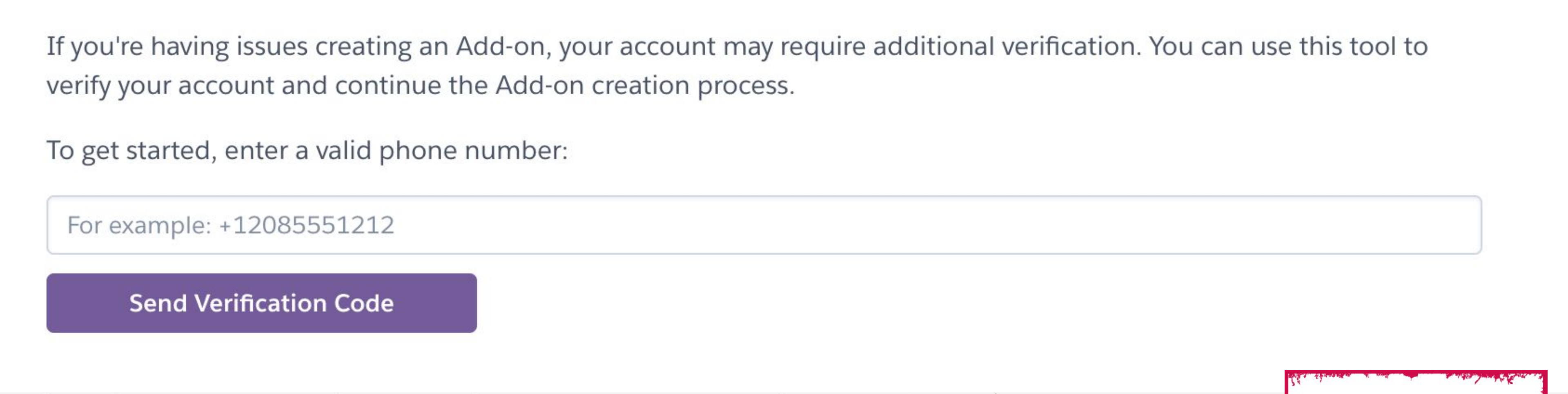
```
$ heroku addons:create heroku-redis:hobby-dev
Creating heroku-redis:hobby-dev on ⬢ rocky-anchorage-37573... !
 ▸   Please verify your account to install this add-on plan (please enter
a credit card) For more information, see
https://devcenter.heroku.com/categories/billing Verify now at
https://heroku.com/verify

$ heroku addons:create heroku-redis:hobby-dev
Creating heroku-redis:hobby-dev on ⬢ rocky-anchorage-37573... free
Your add-on should be available in a few minutes.
! WARNING: Data stored in hobby plans on Heroku Redis are not persisted.
redis-spherical-76470 is being created in the background. The app will
restart when complete...
Use heroku addons:info redis-spherical-76470 to check creation progress
Use heroku addons:docs heroku-redis to view documentation
```

Ustawienie mailgun



```
$ heroku addons:create mailgun:starter

Creating mailgun:starter on ⬢ rocky-anchorage-37573... free
Successfully created your Mailgun account. Happy Sending!
Created mailgun-perpendicular-82936 as MAILGUN_API_KEY, MAILGUN_DOMAIN,
MAILGUN_PUBLIC_KEY, MAILGUN_SMTP_LOGIN, MAILGUN_SMTP_PASSWORD,
MAILGUN_SMTP_PORT, MAILGUN_SMTP_SERVER
Use heroku addons:docs mailgun to view documentation
```

Ustawienie mailgun



If you're having issues creating an Add-on, your account may require additional verification. You can use this tool to verify your account and continue the Add-on creation process.

To get started, enter a valid phone number:

Ustawienie zmiennych

```
$ heroku config:set PYTHONHASHSEED=random
Setting PYTHONHASHSEED and restarting ⬢ rocky-anchorage-37573... done, v5
PYTHONHASHSEED: random

$ heroku config:set WEB_CONCURRENCY=4
Setting WEB_CONCURRENCY and restarting ⬢ rocky-anchorage-37573... done, v6
WEB_CONCURRENCY: 4

$ heroku config:set DJANGO_DEBUG=False
Setting DJANGO_DEBUG and restarting ⬢ rocky-anchorage-37573... done, v7
DJANGO_DEBUG: False
```

Ustawienie zmiennych

```
$ heroku config:set
DJANGO_SETTINGS_MODULE=config.settings.production
Setting DJANGO_SETTINGS_MODULE and restarting ⬢
rocky-anchorage-37573... done, v8
DJANGO_SETTINGS_MODULE: config.settings.production

$ heroku config:set DJANGO_SECRET_KEY="$(openssl rand -base64
64)"
Setting DJANGO_SECRET_KEY and restarting ⬢
rocky-anchorage-37573... done, v9
DJANGO_SECRET_KEY:
jcZsJVFsEjdjGdkbriHKUQkN7ncJt8aA0fg1I2cTQzW3jJmd2xVKpjEB+nonLnFy
j+9ZR3E852HV5d3BAbzF/A==
```

Ustawienie zmiennych

```
$ heroku config:set DJANGO_ADMIN_URL="$(openssl rand -base64
4096 | tr -dc 'A-HJ-NP-Za-km-z2-9' | head -c 32)/"
Setting DJANGO_ADMIN_URL and restarting ⬢
rocky-anchorage-37573... done, v10
DJANGO_ADMIN_URL: iFK8NraWrRwMWc7wNXnBwLheGFH64N4L/
```

Ustawienie zmiennych

```
$ heroku config:set

DJANGO_ALLOWED_HOSTS=rocky-anchorage-37573.herokuapp.com

Setting DJANGO_ALLOWED_HOSTS and restarting ⬢

rocky-anchorage-37573... done, v11

DJANGO_ALLOWED_HOSTS: rocky-anchorage-37573.herokuapp.com
```

Dodanie pliku Procfile

Procfile

```
release: python manage.py migrate
{%- if cookiecutter.use_async == "y" %}
web: gunicorn config.asgi:application -k uvicorn.workers.UvicornWorker
{%- else %}
web: gunicorn config.wsgi:application
{%- endif %}
{%- if cookiecutter.use_celery == "y" %}
worker: REMAP_SIGTERM=SIGQUIT celery -A config.celery_app worker --loglevel=info
beat: REMAP_SIGTERM=SIGQUIT celery -A config.celery_app beat --loglevel=info
{%- endif %}
```

Wysłanie projektu

```
$ git push heroku master

$ heroku run python manage.py createsuperuser
```

PyWWW