



Zaawansowane Techniki Internetowe  
Applikacja: „Budgety”

Projekt został zrealizowany – aplikacja umożliwia zarządzanie budżetem. Wykorzystany framework: Django. Poniżej zamieszczone są jej funkcje oraz wygląd. Poniżej lista rzeczy, które muszą być zainstalowane w celu uruchomienia:

- Python
- Django
- Matplotlib

Github: <https://github.com/SzymonIgor/Budgety/>

W celu uruchomienia servera należy uruchomić terminal w głównym folderze aplikacji i wywołać komendę **python manage.py runserver**. Domyślnie na **localhost:8000** będzie funkcjonowała aplikacja.

```
(projects) C:\Users\szymo\OneDrive\Pulpit\Budgety\budgety_poject>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

You have 2 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): auth.
Run 'python manage.py migrate' to apply them.
May 11, 2020 - 16:20:49
Django version 3.0.6, using settings 'budget_project.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

1. Strona logowania (domyślnie również ustawiona jako strona na którą następuje przekierowania w momencie, gdy użytkownik nie będąc zalogowany próbuje dostać się na stronę inną niż logowania lub tworzenia konta.

Please sign in to view your budget  
informtion.

Username:

Password:

Don't have an account? [Create one here.](#)

```
{% load static %}
<!DOCTYPE html>

<html>

<head>
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css/materialize.min.css">
    <title>Budget App</title>

    <link rel="stylesheet" href="{% static 'style.css' %}">
    <link rel="shortcut icon" href="{% static 'favicon.ico' %}">
    <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
</head>

<body>
    <div class="section">
        <div class="container">
            <h1 class="center">Please sign in to view your budget informtion.</h1>
        </div>
    </div>
    <div class="section">
        <div class="container">
            <form method="post">
                {% csrf_token %}
                {{ form.as_p }}
                <button class="btn waves-effect" type="submit">Login</button>
            </form>
            <br>
            Don't have an account?<a href='{% url "sign up" %}'><strong> Create one here.</strong></a>
        </div>
    </div>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/js/materialize.min.js"></script>
</body>
</html>
```

## 2. Strona do rejestracji – ukazuje się po kliknięciu na odnośnik „Create one here”

# Create an account

Username:

Required. 150 characters or fewer. Letters, digits and @/./+/-/\_ only.

Password:

Your password can't be too similar to your other personal information.

Your password must contain at least 8 characters.

Your password can't be a commonly used password.

Your password can't be entirely numeric.

Password confirmation:

Enter the same password as before, for verification.

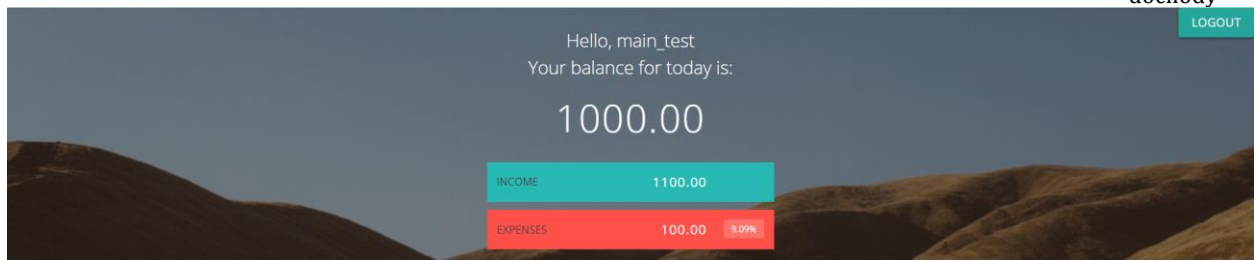
REGISTER

```
{% load static %}
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css/materialize.min.css">

  <link rel="stylesheet" href="{% static 'style.css' %}">
  <link rel="shortcut icon" href="{% static 'favicon.ico' %}">
  <title>Document</title>
</head>
<body>
  <div class="container">
    <h1 class="center">Create an account</h1>
    <form method="post" action = '{% url "sign up" %}'>
      {% csrf_token %}
      {{form.as_p}}
      <button class='btn waves-effect' type="submit">Register</button>
    </form>
  </div>
</body>
</html>
```

3. Główna strona ukazująca się po zalogowaniu

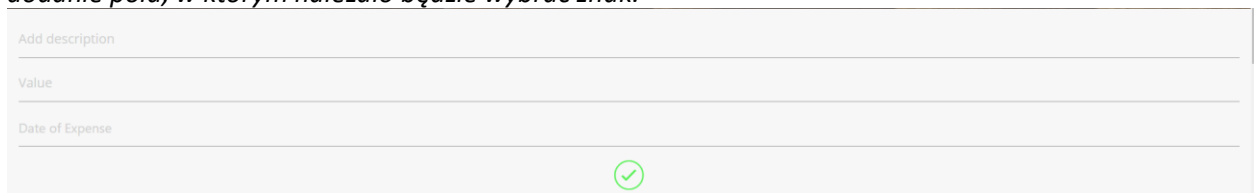
- a. Na samej górze strony znajduje się informacja o aktualnym budżecie, nieco poniżej, sumaryczna ilość dochodów oraz wydatków (przy wydatkach również umieszczona jest ich procentowa wartość –  $\frac{\text{wydatki}}{\text{dochody}}$ ).



```
<div class="top">
  <div class="Logout">
    <ul class="right">
      <li><a href='{% url "logout" %}' class='btn'>Logout</a></li>
    </ul>
  </div>

  <div class="budget">
    <div class="budget__title">
      Hello, {{ user }}
      <br>
      Your balance for today is:
    </div>
    <div class="budget__value">{{ budget_current }}</div>
    <div class="budget__income clearfix">
      <div class="budget__income--text">Income</div>
      <div class="right">
        <div class="budget__income--value">{{ income }}</div>
        <div class="budget__income--percentage">{{ income_prc }}</div>
      </div>
    </div>
    <div class="budget__expenses clearfix">
      <div class="budget__expenses--text">Expenses</div>
      <div class="right">
        <div class="budget__expenses--value">{{ expenses }}</div>
        <div class="budget__expenses--percentage">{{ expenses_prc }}</div>
      </div>
    </div>
  </div>
</div>
```

- b. Poniżej znajduje się miejsce, w którym można dodawać pojedyncze płatności / obciążenia / zarobki. Aby dodać dochód należy wpisać liczbę dodatnią, natomiast wydatek – liczbę ze znakiem „-„. *Może być to nieintuicyjne i zapewne istnieje możliwość poprawienia tego w łatwy sposób – na przykład poprzez dodanie pola, w którym należało będzie wybrać znak.*



The form is titled 'Add description' and has three input fields: 'Value', 'Date of Expense', and a green checkmark icon.

```
<div class="add">
  <div class="add__container">
    <form method='post' action = '{% url "add item" %}'>
      {% csrf_token %}
      {{ form.as_p }}
      <input type="text" class="add__description" placeholder="Add description" name="expense_name"
        id="expense_name">
      <input type="number" class="add__value" placeholder="Value" name="cost" id="cost">
      <input type="text" class="datepicker" placeholder="Date of Expense" name="date_added" id="date_added">
      <button class="add__btn"><i class="ion-ios-checkmark-outline"></i></button>
    </form>
  </div>
</div>
```

Wszystkie pola są wymagane. Jeżeli któreś z nich nie zostanie wypełnione na stronie pojawi się komunikat. Prawdopodobnie okienko pop-up mogłoby się lepiej sprawdzić.

Please fill up all of the fields below to add an item

Add description

Value

Date of Expense

✓

```

<div class="messages">
  {% if messages %}
    <ul class="messages">
      {% for message in messages %}
        <li class="{{ message.tags }}">{{ message }}</li>
      {% endfor %}
    </ul>
  {% endif %}
</div>

```

Po kliknięciu w pole daty wyświetli się interaktywny kalendarz – umożliwiający wybranie daty dodawanego rekordu.

Add description

value

Date of Expense

2020

Mon, May 11

<

May

2020

>

S

M

T

W

T

F

S

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

CANCEL

OK

600

400

200

0

expenses

income

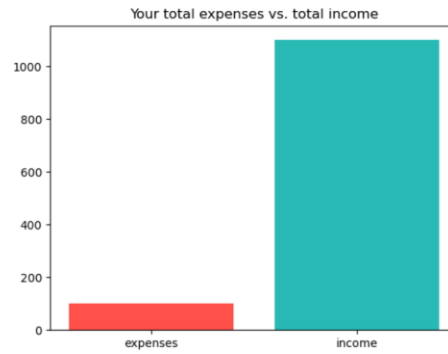
- c. Poniżej znajduje się sumaryczny wykres – generowany na bieżąco - oraz wszystkie dodane rzeczy (uporządkowane od najmlodszeo).



```



```



#### EXPENSES

May 3, 2020	expense_1	-100.00
-------------	-----------	---------

#### INCOMES

May 8, 2020	income_1	100.00
May 1, 2020	income_2	1000.00

*\*analogicznie dla Incomes*

```
<div class="container clearfix">
  <div class="expenses"> <h2 class="expenses_title">Expenses</h2>
  {% for expense_item in expense_items %}
  {% if expense_item.cost < 0 %}

    <div class="item clearfix">
      <div class="item__date">{{ expense_item.date_added }}</div>
      <div class="item__description">{{ expense_item.expense_name }}</div>

      <div class="right">
        <div class="item__value">{{ expense_item.cost }}</div>
        <div class="item__delete">
          <form method="post" action="{% url 'delete item' %}">
            {% csrf_token %}
            {{ form.as_p }}
            <input type="hidden" class="id" placeholder="Enter ID" name="id" id="id" value="{{ expense_item.id }}">
            <button class="item__delete--btn"><i class="ion-ios-close-outline"></i></button>
          </form>
        </div>
      </div>
    </div>
  {% endif %}
  {% endfor %}
</div>
```

Istnieje możliwość usuwania poszczególnych rekordów. Po najechaniu na wiersz, który planujemy usunąć kwota przesunie się w lewo a w jej miejscu pojawi się krzyżyk – po jego naciśnięciu rekord zostanie usunięty, a wszystkie dane zaktualizowane.

#### INCOMES

May 8, 2020	income_1	100.00	✕
-------------	----------	--------	---

#### INCOMES

May 8, 2020	income_1	100.00	✕
-------------	----------	--------	---

#### INCOMES

May 8, 2020	income_1	100.00	✕
-------------	----------	--------	---

```
.item__delete--btn:focus { outline: none; }
.item__delete--btn:active { transform: translateY(2px); }

.item:hover .item__delete--btn { display: block; }
.item:hover .item__value { transform: translateX(-25px); }
```

4. Istnieje możliwość dodawania nowych jak i usuwania już istniejących użytkowników poprzez panel administratora. Aby z niego skorzystać należy być zalogowanym jako administrator (login: admin, hasło: admin) oraz wpisać w adresie strony localhost:8000/admin.

Jeśli nie będziemy zalogowani jako administrator zostaniemy przekierowani na poniższą stronę, która o tym poinformuje oraz umożliwi administratorowi zalogowanie się.

Po zalogowaniu na konto z uprawnieniami jest możliwość wykonywania akcji wcześniej opisanych.

5. Plan początkowy został zrealizowany, stworzono również niektóre z funkcjonalności dodatkowych.

Plan	Czy zrealizowano?
tworzenie kont	Tak
logowanie	Tak
możliwość wylogowania	Tak
panel administratora do zarządzania użytkownikami	Tak
blokada podstron podczas bycia niezalogowanym	Tak
przekierowanie związane z blokadą	Tak
dodawanie rekordów	Tak
obsługa wyjątków - pola są niewypełnione (lub wypełnione nieprawidłowo)	Tak
interaktywny kalendarz umożliwiający wprowadzanie daty	Tak
przyjazny dla użytkownika wygląd	Tak
wyświetlanie rekordów względem ich dat	Tak
usuwanie wybranych rekordów	Tak
<b>Dodatkowo:</b> interaktywne pole do usuwania (ruchomy tekst po najechnięciu myszką i pokazujący się przycisk)	Tak
<b>Dodatkowo:</b> wykres generowany na bieżąco	Tak
<b>Dodatkowo:</b> osobne pole znaku dla bardziej intuicyjnego rozróżnienia dochodów i wydatków (zamiast znaku przy wpisywanej liczbie)	Nie
<b>Dodatkowo:</b> możliwość edycji rekordów (zamiast usuwania i dodawania na nowo)	Nie



## 6. Back-end

### views.py

```
@login_required(login_url='/')
def index(request):
    expense_items = ExpenseInfo.objects.filter(user_expense=request.user).order_by('-date_added')
    try:
        income_total = ExpenseInfo.objects.filter(user_expense=request.user).\
            aggregate(income=Sum('cost', filter=Q(cost__gt=0)))
        expense_total = ExpenseInfo.objects.filter(user_expense=request.user).\
            aggregate(expenses=Sum('cost', filter=Q(cost__lt=0)))

        except TypeError:
            print('No data.')
        expenses = round(abs(expense_total['expenses']) if expense_total['expenses'] is not None else 0, 2)
        income = round(abs(income_total['income']) if income_total['income'] is not None else 0, 2)
        budget_current = round(income - expenses, 2)
        fig, ax = plt.subplots()
        ax.bar(['expenses', 'income'], [expenses, income], color=['#FF5049', '#28B9B5'])
        ax.set_title('Your total expenses vs. total income')
        plt.savefig('budget_app/static/expense.png')
        try:
            expenses_prc = round(expenses/income * 100, 2)
        except:
            expenses_prc = 0
        context = {'expense_items': expense_items, 'income': income, 'expenses': expenses, 'budget_current': budget_current,
                   'expenses_prc': expenses_prc}
        return render(request, 'budget_app/index.html', context=context)

@login_required(login_url='/')
def add_item(request):
    try:
        name = request.POST['expense_name']
        expense_cost = request.POST['cost']
        date_added = request.POST['date_added']
        ExpenseInfo.objects.create(expense_name=name, cost=expense_cost, date_added=date_added,
                                   user_expense=request.user)
    except:
        messages.error(request, "Please fill up all of the fields below to add an item")
    return HttpResponseRedirect('app')

@login_required(login_url='/')
def del_item(request):
    id = request.POST['id']
    ExpenseInfo.objects.filter(id=id).delete()
    return HttpResponseRedirect('app')

@login_required(login_url='/')
def logout_view(request):
    logout(request)
    return HttpResponseRedirect('/')

def sign_up(request):
    if request.method == 'POST':
        form = UserCreationForm(request.POST)
        if form.is_valid():
            user = form.save()
            login(request, user)
            return HttpResponseRedirect('app')
        else:
            for msg in form.error_messages:
                print(form.error_messages[msg])
    else:
        form = UserCreationForm
        return render(request, 'budget_app/sign_up.html', {'form': form})
```

## models.py

```
from django.db import models
from django.utils import timezone
from django.contrib.auth.models import User

# Create your models here.

class ExpenseInfo(models.Model):
    id = models.AutoField(primary_key=True)
    expense_name = models.CharField(max_length=20)
    cost = models.DecimalField(decimal_places=2, max_digits=1_000_000_000)
    date_added = models.DateField()
    user_expense = models.ForeignKey(User, on_delete=models.CASCADE)
```

## urls.py

```
from django.urls import path, include
from . import views
from django.contrib.auth import views as auth_views

urlpatterns = [
    path('', auth_views.LoginView.as_view(), name='login'),
    path('app', views.index, name='index'),
    path('add_item', views.add_item, name='add item'),
    path('del_item', views.del_item, name='delete item'),
    path('logout', views.logout_view, name='logout'),
    path('sign_up', views.sign_up, name="sign up")
]
```

## 7. Główna strona w całości

