**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ**

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Департамент программной инженерии

|  |  |  |
| --- | --- | --- |
| СОГЛАСОВАНО  Стажер-исследователь научной лаборатории моделей и методов вычислительной прагматики  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д. А. Киселев  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г. |  | УТВЕРЖДАЮ  Академический руководитель образовательной программы «Программная инженерия»  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.В. Шилов  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г. |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Подп. и дата |  | | Инв. № дубл. |  | | Взам. Инв. № |  | | Подп. и дата |  | | Инв. № подл. |  | | **ИНТЕРАКТИВНЫЕ ДАШБОРДЫ ДЛЯ АНАЛИТИКИ ТРАЕКТОРИЙ ПОЛЬЗОВАТЕЛЕЙ   Текст программы**  **ЛИСТ УТВЕРЖДЕНИЯ**  **RU.17701729.04.13 12 01-1-ЛУ**  Исполнитель  Студент группы БПИ172  \_\_\_\_\_\_\_\_\_\_\_\_/А.Н. Карпин/  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г. | |
|  |  |

**УТВЕРЖДЕН**

**RU.17701729.04.13 12 01-1-ЛУ**

|  |  |
| --- | --- |
| Подп. и дата |  |
| Инв. № дубл. |  |
| Взам. инв. № |  |
| Подп. и дата |  |
| Инв. № подл |  |

**ИНТЕРАКТИВНЫЕ ДАШБОРДЫ ДЛЯ АНАЛИТИКИ ТРАЕКТОРИЙ ПОЛЬЗОВАТЕЛЕЙ**

**Текст программы  
  
RU.17701729.04.13 12 01-1**

**Листов 73**

**Содержание**

[**ТЕКСТ ПРОГРАММНОГО КОМПЛЕКСА** 3](#_Toc37602707)

[**1.** **Текст программного комплекса** 3](#_Toc37602708)

[**1.1** **Файл retentionboards\_core/app/retentionboards\_core/settings.py** 3](#_Toc37602709)

[**1.2** **Файл retentionboards\_core/app/retentionboards\_core/urls.py** 9](#_Toc37602710)

[**1.3** **Файл retentionboards\_core/app/retentionboards\_core/wsgi.py** 10](#_Toc37602711)

[**1.4** **Файл retentionboards\_core/app/retentionboards\_core/\_\_init\_\_.py** 11](#_Toc37602712)

[**1.5** **Файл retentionboards\_core/app/retentionboards\_core/asgi.py** 11](#_Toc37602713)

[**1.6** **Файл retentionboards\_core/app/dashboards/views.py** 12](#_Toc37602714)

[**1.7** **Файл retentionboards\_core/app/dashboards/urls.py** 18](#_Toc37602715)

[**1.8** **Файл retentionboards\_core/app/dashboards/models.py** 18](#_Toc37602716)

[**1.9** **Файл retentionboards\_core/app/dashboards/forms.py** 20](#_Toc37602717)

[**1.10** **Файл retentionboards\_core/app/users/views.py** 21](#_Toc37602718)

[**1.11** **Файл retentionboards\_core/app/users/urls.py** 24](#_Toc37602719)

[**1.12** **Файл retentionboards\_core/app/users/models.py** 24](#_Toc37602720)

[**1.13** **Файл retentionboards\_core/app/users/forms.py** 25](#_Toc37602721)

[**1.14** **Файл retentionboards\_core/app/users/management/commands/initadmin.py** 26](#_Toc37602722)

[**1.15** **Файл retentionboards\_core/app/celery\_manager.py** 26](#_Toc37602723)

[**1.16** **Файл retentionboards\_core/app/Dockerfile** 27](#_Toc37602724)

[**1.17** **Файл retentionboards\_core/app/Dockerfile.prod** 28](#_Toc37602725)

[**1.18** **Файл retentionboards\_core/app/entrypoint.sh** 31](#_Toc37602726)

[**1.19** **Файл retentionboards\_core/app/entrypoint.prod.sh** 32](#_Toc37602727)

[**1.20** **Файл retentionboards\_core/app/manage.py** 32](#_Toc37602728)

[**1.21** **Файл retentionboards\_core/app/templates/base.html** 33](#_Toc37602729)

[**1.22** **Файл retentionboards\_core/app/templates/dashboard.html** 42](#_Toc37602730)

[**1.23** **Файл retentionboards\_core/app/templates/experiment.html** 45](#_Toc37602731)

[**1.24** **Файл retentionboards\_core/app/templates/graph.html** 51](#_Toc37602732)

[**1.25** **Файл retentionboards\_core/app/templates/index.html** 55](#_Toc37602733)

[**1.26** **Файл retentionboards\_core/nginx/Dockerfile** 57](#_Toc37602734)

[**1.27** **Файл retentionboards\_core/nginx/nginx.conf** 58](#_Toc37602735)

[**1.28** **Файл docker-compose.prod.yml** 59](#_Toc37602736)

[**1.29** **Файл docker-compose.yml** 62](#_Toc37602737)

[**1.30** **Файл retentionboards\_retentioneering/app/analytics/tasks.py** 66](#_Toc37602738)

[**1.31** **Файл retentionboards\_retentioneering/app/analytics/redis\_connector.py** 70](#_Toc37602739)

[**ТЕРМИНОЛОГИЯ** 71](#_Toc37602741)

[**СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ** 72](#_Toc37602743)

[**ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ** 2](#_Toc37602744)

# **ТЕКСТ ПРОГРАММНОГО КОМПЛЕКСА**

# **Текст программного комплекса**

## **Файл retentionboards\_core/app/retentionboards\_core/settings.py**

*"""*

*Django settings for RetentionBoards project.*

*Generated by 'django-admin startproject' using Django 3.0.4.*

*For more information on this file, see*

*https://docs.djangoproject.com/en/3.0/topics/settings/*

*For the full list of settings and their values, see*

*https://docs.djangoproject.com/en/3.0/ref/settings/*

*"""*

*import os*

*from celery.schedules import crontab*

*# Build paths inside the project like this: os.path.join(BASE\_DIR, ...)*

*BASE\_DIR = os.path.dirname(os.path.dirname(os.path.abspath(\_\_file\_\_)))*

*# Quick-start development settings - unsuitable for production*

*# See https://docs.djangoproject.com/en/3.0/howto/deployment/checklist/*

*# SECURITY WARNING: keep the secret key used in production secret!*

*SECRET\_KEY = os.environ.get("SECRET\_KEY", default="some key")*

*# SECURITY WARNING: don't run with debug turned on in production!*

*DEBUG = int(os.environ.get("DEBUG", default=0))*

*ALLOWED\_HOSTS = ["\*"]*

*# Application definition*

*INSTALLED\_APPS = [*

*'django.contrib.admin',*

*'django.contrib.auth',*

*'django.contrib.contenttypes',*

*'django.contrib.sessions',*

*'django.contrib.messages',*

*'django.contrib.staticfiles',*

*'users',*

*'dashboards',*

*'django\_celery\_beat',*

*#'guardian',*

*]*

*MIDDLEWARE = [*

*'django.middleware.security.SecurityMiddleware',*

*'django.contrib.sessions.middleware.SessionMiddleware',*

*'django.middleware.common.CommonMiddleware',*

*'django.middleware.csrf.CsrfViewMiddleware',*

*'django.contrib.auth.middleware.AuthenticationMiddleware',*

*'django.contrib.messages.middleware.MessageMiddleware',*

*'django.middleware.clickjacking.XFrameOptionsMiddleware',*

*]*

*ROOT\_URLCONF = 'retentionboards\_core.urls'*

*TEMPLATES = [*

*{*

*'BACKEND': 'django.template.backends.django.DjangoTemplates',*

*'DIRS': [os.path.join(BASE\_DIR, 'templates')]*

*,*

*'APP\_DIRS': True,*

*'OPTIONS': {*

*'context\_processors': [*

*'django.template.context\_processors.debug',*

*'django.template.context\_processors.request',*

*'django.contrib.auth.context\_processors.auth',*

*'django.contrib.messages.context\_processors.messages',*

*],*

*},*

*},*

*]*

*WSGI\_APPLICATION = 'retentionboards\_core.wsgi.application'*

*# Database*

*# https://docs.djangoproject.com/en/3.0/ref/settings/#databases*

*DATABASES = {*

*"default": {*

*"ENGINE": os.environ.get("SQL\_ENGINE", "django.db.backends.sqlite3"),*

*"NAME": os.environ.get("SQL\_DATABASE", os.path.join(BASE\_DIR, "db.sqlite3")),*

*"USER": os.environ.get("SQL\_USER", "user"),*

*"PASSWORD": os.environ.get("SQL\_PASSWORD", "password"),*

*"HOST": os.environ.get("SQL\_HOST", "localhost"),*

*"PORT": os.environ.get("SQL\_PORT", "5432"),*

*}*

*}*

*CACHES = {*

*"default": {*

*"BACKEND": os.environ.get("CACHE\_ENGINE", "django.core.cache.backends.memcached.MemcachedCache"),*

*"LOCATION": os.environ.get("CACHE\_LOCATION", "127.0.0.1:11211"),*

*"OPTIONS": {*

*"CLIENT\_CLASS": os.environ.get("CACHE\_CLIENT\_CLASS", ""),*

*}*

*}*

*}*

*# Password validation*

*# https://docs.djangoproject.com/en/3.0/ref/settings/#auth-password-validators*

*AUTH\_PASSWORD\_VALIDATORS = [*

*{*

*'NAME': 'django.contrib.auth.password\_validation.UserAttributeSimilarityValidator',*

*},*

*{*

*'NAME': 'django.contrib.auth.password\_validation.MinimumLengthValidator',*

*},*

*{*

*'NAME': 'django.contrib.auth.password\_validation.CommonPasswordValidator',*

*},*

*{*

*'NAME': 'django.contrib.auth.password\_validation.NumericPasswordValidator',*

*},*

*]*

*LOGGING = {*

*'version': 1,*

*'disable\_existing\_loggers': False,*

*'handlers': {*

*'file\_django': {*

*'level': 'DEBUG',*

*'class': 'logging.FileHandler',*

*'filename': 'logs/logger.log',*

*},*

*'file\_django\_requests': {*

*'level': 'DEBUG',*

*'class': 'logging.FileHandler',*

*'filename': 'logs/request\_logger.log',*

*}*

*},*

*'loggers': {*

*'django': {*

*'handlers': ['file\_django'],*

*'level': 'DEBUG',*

*'propagate': True,*

*},*

*'django.request': {*

*'handlers': ['file\_django\_requests'],*

*'level': 'DEBUG',*

*'propagate': False,*

*}*

*},*

*}*

*# Internationalization*

*# https://docs.djangoproject.com/en/3.0/topics/i18n/*

*LANGUAGE\_CODE = 'en-us'*

*TIME\_ZONE = 'UTC'*

*USE\_I18N = True*

*USE\_L10N = True*

*USE\_TZ = True*

*# Static files (CSS, JavaScript, Images)*

*# https://docs.djangoproject.com/en/3.0/howto/static-files/*

*STATICFILES\_DIRS = ( os.path.join('staticfiles'), )*

*STATIC\_URL = '/staticfiles/'*

*STATIC\_ROOT = '/staticfiles/'*

*MEDIA\_URL = '/mediafiles/'*

*MEDIA\_ROOT = os.path.join(BASE\_DIR, 'mediafiles')*

*CELERY\_BROKER\_URL = os.environ.get("CELERY\_BROKER\_URL", 'redis://redis\_queue:6379/0')*

*CELERY\_RESULT\_BACKEND = os.environ.get("CELERY\_RESULT\_BACKEND", 'redis://redis\_queue:6379/')*

*CELERY\_ACCEPT\_CONTENT = ['application/json']*

*CELERY\_RESULT\_SERIALIZER = 'json'*

*CELERY\_TASK\_SERIALIZER = 'json'*

*CELERY\_TIMEZONE = 'Europe/Moscow'*

## **Файл retentionboards\_core/app/retentionboards\_core/urls.py**

**"""RetentionBoards URL Configuration**

**The `urlpatterns` list routes URLs to views. For more information please see:**

**https://docs.djangoproject.com/en/3.0/topics/http/urls/**

**Examples:**

**Function views**

**1. Add an import: from my\_app import views**

**2. Add a URL to urlpatterns: path('', views.home, name='home')**

**Class-based views**

**1. Add an import: from other\_app.views import Home**

**2. Add a URL to urlpatterns: path('', Home.as\_view(), name='home')**

**Including another URLconf**

**1. Import the include() function: from django.urls import include, path**

**2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))**

**"""**

**from django.contrib import admin**

**from django.urls import path, include**

**import users.urls**

**import dashboards.urls**

**urlpatterns = [**

**path('admin/', admin.site.urls),**

**path('web/app/', include(users.urls)),**

**path('web/app/dashboard/', include(dashboards.urls))**

**]**

## **Файл retentionboards\_core/app/retentionboards\_core/wsgi.py**

*"""*

*WSGI config for RetentionBoards project.*

*It exposes the WSGI callable as a module-level variable named ``application``.*

*For more information on this file, see*

*https://docs.djangoproject.com/en/3.0/howto/deployment/wsgi/*

*"""*

*import os*

*from django.core.wsgi import get\_wsgi\_application*

*os.environ.setdefault('DJANGO\_SETTINGS\_MODULE', 'retentionboards\_core.settings')*

*application = get\_wsgi\_application()*

## **Файл retentionboards\_core/app/retentionboards\_core/\_\_init\_\_.py**

from \_\_future\_\_ import absolute\_import, unicode\_literals

from celery\_manager import app as celery\_manager

\_\_all\_\_ = ('celery\_manager',)

## **Файл retentionboards\_core/app/retentionboards\_core/asgi.py**

"""

ASGI config for RetentionBoards project.

It exposes the ASGI callable as a module-level variable named ``application``.

For more information on this file, see

https://docs.djangoproject.com/en/3.0/howto/deployment/asgi/

"""

import os

from django.core.asgi import get\_asgi\_application

os.environ.setdefault('DJANGO\_SETTINGS\_MODULE', 'retentionboards\_core.settings')

application = get\_asgi\_application()

## **Файл retentionboards\_core/app/dashboards/views.py**

from django.http import HttpResponse

from django.shortcuts import render, redirect, get\_object\_or\_404

from django.views import View

from django.contrib import messages

from django.views.decorators.csrf import csrf\_exempt

from dashboards.models import EventSet, Event, EventsetStatus

from dashboards.forms import UploadForm

from celery.result import AsyncResult

from celery\_manager import app as celery\_app

from redis import Redis

import csv, io

import logging

import json

import os

import pickle

from PIL import Image

logger = logging.getLogger('django')

request\_logger = logging.getLogger('django.request')

class EventsetsView(View):

template = "dashboard.html"

def get(self, request):

if request.user.is\_authenticated:

form = UploadForm()

eventsets\_list = EventSet.get\_eventset\_list(user=request.user)

return render(request, self.template, {'eventsets': eventsets\_list, 'form': form})

else:

messages.error(request, 'USER IS NOT AUTHENTICATED')

return redirect('/web/app/')

def post(self, request):

if request.user.is\_authenticated:

form = UploadForm(request.POST, request.FILES)

if form.is\_valid():

csv\_file = request.FILES['file']

if not csv\_file.name.endswith('.csv'):

messages.error(request, 'THIS IS NOT A CSV FILE')

data\_set = csv\_file.read().decode('UTF-8')

io\_string = io.StringIO(data\_set)

next(io\_string)

eventset, \_ = EventSet.objects.update\_or\_create(name=form.data['name'], user=request.user,

status=EventsetStatus.NEW)

print(eventset.id)

event\_list = []

users = set()

unique\_events = set()

for column in csv.reader(io\_string, delimiter=',', quotechar="|"):

users.add(column[2])

unique\_events.add(column[0])

event = Event(

event\_name=column[0],

event\_timestamp=column[1],

user\_pseudo\_id=column[2],

event\_set=eventset

)

event\_list.append(event)

eventset.events = len(event\_list)

eventset.users = len(users)

eventset.unique\_events=len(unique\_events)

Event.objects.bulk\_create(event\_list)

task\_id = celery\_app.send\_task(name='prepare\_dataset', args=[eventset.id], queue='retention\_queue\_hi')

task\_tsne = celery\_app.send\_task(name='learn\_tsne', args=[eventset.id], queue='retention\_queue\_hi')

eventset.step\_matrix\_task\_id = task\_id

eventset.tsne\_task\_id = task\_tsne

eventset.save()

eventsets\_list = EventSet.get\_eventset\_list(user=request.user)

form = UploadForm()

return render(request, self.template, {'eventsets': eventsets\_list, 'task\_id': task\_id, 'task\_tsne':

task\_tsne, 'eventset\_name': eventset.name, 'form': form})

else:

messages.error(request, 'FORM IS NOT VALID')

return redirect('/web/app/dashboard/')

else:

messages.error(request, 'USER IS NOT AUTHENTICATED')

return redirect('/web/app/')

class ExperimentView(View):

template = 'experiment.html'

def get(self, request, pk):

request\_logger.debug(request)

if request.user.is\_authenticated:

#if request.user.has\_perm('crud portfolio', portfolio):

redis = Redis(host=os.environ.get("REDIS\_HOST", 'redis'), port=os.environ.get("REDIS\_PORT", 6379))

eventset = get\_object\_or\_404(EventSet, pk=pk)

task\_id = eventset.step\_matrix\_task\_id

task\_tsne = eventset.tsne\_task\_id

step\_matrix\_link = f'images/experiments/step\_matrix/file{pk}.png'

if AsyncResult(task\_id).result is not None:

step\_matrix = pickle.loads(redis.get('heatmap' + str(pk)))

step\_matrix.save('staticfiles/' + step\_matrix\_link)

tsne\_link = f'images/experiments/tsne/file{pk}.png'

if AsyncResult(task\_tsne).result is not None:

tsne = pickle.loads(redis.get('tsne' + str(pk)))

tsne.save('staticfiles/' + tsne\_link)

graph = None

graphs\_all = []

if AsyncResult(task\_tsne).result is not None:

for cluster in AsyncResult(task\_tsne).result[1]:

graph = pickle.loads(redis.get(cluster))

with open(f'staticfiles/images/experiments/graphs/{cluster}.html', 'w+') as f:

f.write(graph)

graphs\_all.append([f'/web/app/dashboard/experiments/{pk}/{cluster}/', cluster])

return render(request, self.template, {'eventset': eventset, 'tsne\_link': tsne\_link,

'step\_matrix\_link': step\_matrix\_link, 'graphs\_all': graphs\_all})

#else:

# portfolios = Portfolio.objects.filter(user=request.user.profile).all()

# portfolio\_list = get\_portfolios\_list(portfolios=portfolios)

# messages.add\_message(request, messages.ERROR, 'You do not have access to this portfolio')

# return render(request, 'portfolios\_list.html', {'list': portfolio\_list})

else:

messages.add\_message(request, messages.ERROR, 'You are not authentificated')

return redirect('/web/app/')

class GraphView(View):

template = "graph.html"

def get(self, request, pk, cluster):

if request.user.is\_authenticated:

with open(f'staticfiles/images/experiments/graphs/{cluster}.html', 'r') as f:

graph = f.read()

return render(request, self.template, {'graph': graph})

else:

messages.error(request, 'USER IS NOT AUTHENTICATED')

return redirect('/web/app/')

@csrf\_exempt

def check\_task\_status(request):

if request.is\_ajax():

if 'task\_id' in request.POST.keys() and request.POST['task\_id']:

task\_id = request.POST['task\_id']

task = AsyncResult(task\_id)

data = task.state

else:

data = 'No task\_id in the request'

else:

data = 'This is not an ajax request'

json\_data = json.dumps(data)

return HttpResponse(json\_data, content\_type='application/json')

## **Файл retentionboards\_core/app/dashboards/urls.py**

"""RetentionBoards URL Configuration

The `urlpatterns` list routes URLs to views. For more information please see:

https://docs.djangoproject.com/en/3.0/topics/http/urls/

Examples:

Function views

1. Add an import: from my\_app import views

2. Add a URL to urlpatterns: path('', views.home, name='home')

Class-based views

1. Add an import: from other\_app.views import Home

2. Add a URL to urlpatterns: path('', Home.as\_view(), name='home')

Including another URLconf

1. Import the include() function: from django.urls import include, path

2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))

"""

from django.urls import path

from .views import EventsetsView, check\_task\_status, ExperimentView, GraphView

app\_name = 'dashboards'

urlpatterns = [

path('', EventsetsView.as\_view()),

path('check\_status/', check\_task\_status),

path('experiments/<int:pk>/', ExperimentView.as\_view()),

path('experiments/<int:pk>/<str:cluster>/', GraphView.as\_view()),

path('csv/', EventsetsView.as\_view()),

]

## **Файл retentionboards\_core/app/dashboards/models.py**

from django.db import models

from users.models import User

from enum import Enum

class EventsetStatus(Enum):

SUCCESS = "Success"

PENDING = "Pending"

FAILED = "Failed"

PROGRESS = "Progress"

NEW = "New"

class EventSet(models.Model):

name = models.CharField(max\_length=150)

user = models.ForeignKey(User, on\_delete=models.CASCADE)

status = models.CharField(max\_length=120,

choices=[(tag, tag.value) for tag in EventsetStatus]

)

step\_matrix\_task\_id = models.CharField(max\_length=120)

tsne\_task\_id = models.CharField(max\_length=120)

events = models.IntegerField(null=True)

users = models.IntegerField(null=True)

unique\_events = models.IntegerField(null=True)

@property

def processing(self):

return self.status != EventsetStatus.SUCCESS

@property

def events\_per\_user(self):

return "{0:.2f}".format(self.events/self.users)

@property

def href(self):

return "/web/app/dashboard/experiments/" + str(self.id)

@classmethod

def get\_eventset\_list(cls, user):

eventsets = EventSet.objects.filter(user=user)

eventsets\_list = []

first = 0

counter = 0

row = 0

eventsets\_list.append([])

for eventset in eventsets:

if first < 2:

eventsets\_list[0].append(eventset)

first += 1

if first == 2:

counter = 3

first += 1

continue

if first > 2:

if counter == 3:

eventsets\_list.append([])

row += 1

counter = 0

eventsets\_list[row].append(eventset)

counter += 1

return eventsets\_list

class Event(models.Model):

event\_name = models.CharField(max\_length=150)

event\_timestamp = models.CharField(max\_length=150)

user\_pseudo\_id = models.CharField(max\_length=150)

event\_set = models.ForeignKey(EventSet, on\_delete=models.CASCADE)

## **Файл retentionboards\_core/app/dashboards/forms.py**

from django import forms

class UploadForm(forms.Form):

name = forms.CharField(label='Eventset name', max\_length=100)

file = forms.FileField(label='File')

## **Файл retentionboards\_core/app/users/views.py**

from django.shortcuts import render, redirect

from django.views import View

from users.forms import LoginForm, RegisterForm

from django.contrib.auth import authenticate, logout, login

from django.contrib.auth.models import User

from django.contrib import messages

import logging

logger = logging.getLogger('django')

request\_logger = logging.getLogger('django.request')

class MainView(View):

template\_name = 'index.html'

def get(self, request):

request\_logger.debug(request)

form = LoginForm()

register = RegisterForm()

return render(request, self.template\_name, {'form': form, 'form\_register': register})

class LoginView(View):

def post(self, request):

request\_logger.debug(request)

form = LoginForm(request.POST)

if form.is\_valid():

user = authenticate(username=form.data['username'], password=form.data['password'])

if user is not None:

login(request, user, backend='django.contrib.auth.backends.ModelBackend')

return redirect('/web/app/dashboard/')

else:

form = LoginForm()

register = RegisterForm()

messages.add\_message(request, messages.ERROR, 'Check your login and password')

return render(request, 'index.html', {'form': form, 'form\_register': register})

else:

form = LoginForm()

register = RegisterForm()

messages.add\_message(request, messages.ERROR, 'Check your form data')

return render(request, 'index.html', {'form': form, 'form\_register': register})

class RegisterView(View):

def post(self, request):

request\_logger.debug(request)

form = RegisterForm(request.POST)

if form.is\_valid():

if form.data['password'] == form.data['password2']:

us = User.objects.filter(username=form.data['username']).first()

if us is None:

user = User.objects.create\_user(username=form.data['username'], password=form.data['password'])

user.save()

if user is not None:

login(request, user, backend='django.contrib.auth.backends.ModelBackend')

return redirect('/web/app/dashboard/')

else:

register = RegisterForm()

form = LoginForm()

messages.add\_message(request, messages.ERROR, 'User with this username already exists')

return render(request, 'index.html', {'form': form, 'form\_register': register})

else:

register = RegisterForm()

form = LoginForm()

messages.add\_message(request, messages.ERROR, 'Passwords does not match')

return render(request, 'index.html', {'form': form, 'form\_register': register})

else:

register = RegisterForm()

form = LoginForm()

messages.add\_message(request, messages.ERROR, 'Check your form data')

return render(request, 'index.html', {'form': form, 'form\_register': register})

class LogoutView(View):

def get(self, request):

request\_logger.debug(request)

logout(request)

form = LoginForm()

register = RegisterForm()

return redirect('/web/app/', {'form': form, 'form\_register': register})

## **Файл retentionboards\_core/app/users/urls.py**

from django.contrib import admin

from django.urls import path

from django.conf.urls import include, url

from .views import LoginView, LogoutView, RegisterView, MainView

urlpatterns = [

path('', MainView.as\_view()),

path('login/', LoginView.as\_view()),

path('register/', RegisterView.as\_view()),

path('logout/', LogoutView.as\_view()),

]

## **Файл retentionboards\_core/app/users/models.py**

from django.db import models

from django.contrib.auth.models import User

from django.db.models.signals import post\_save

from django.dispatch import receiver

# Create your models here.

class Profile(models.Model):

user = models.OneToOneField(User, on\_delete=models.CASCADE)

@receiver(post\_save, sender=User)

def create\_user\_profile(sender, instance, created, \*\*kwargs):

if created:

Profile.objects.create(user=instance)

@receiver(post\_save, sender=User)

def save\_user\_profile(sender, instance, \*\*kwargs):

instance.profile.save()

## **Файл retentionboards\_core/app/users/forms.py**

from django import forms

class LoginForm(forms.Form):

username = forms.CharField(label='Username', max\_length=100)

password = forms.CharField(widget=forms.PasswordInput(), label='Password', max\_length=100)

class RegisterForm(forms.Form):

username = forms.CharField(label='Username', max\_length=100)

password = forms.CharField(widget=forms.PasswordInput(), label='Password', max\_length=100)

password2 = forms.CharField(widget=forms.PasswordInput(), label='Password', max\_length=100)

## **Файл retentionboards\_core/app/users/management/commands/initadmin.py**

from django.core.management.base import BaseCommand

from django.contrib.auth.management.commands.createsuperuser import get\_user\_model

import os

class Command(BaseCommand):

def handle(self, \*args, \*\*options):

username = os.getenv('DJANGO\_SU\_NAME', 'admin')

email = os.getenv('DJANGO\_SU\_EMAIL', 'admin@mysite.com')

password = os.getenv('DJANGO\_SU\_PASSWORD', 'mypass')

get\_user\_model().\_default\_manager.db\_manager('default') \

.create\_superuser(username=username, email=email, password=password)

self.stdout.write("SuperUser Created\n", ending='')

## **Файл retentionboards\_core/app/celery\_manager.py**

from \_\_future\_\_ import absolute\_import, unicode\_literals

import os

from celery import Celery

from kombu import Exchange, Queue

os.environ.setdefault('DJANGO\_SETTINGS\_MODULE', 'retentionboards\_core.settings')

app = Celery('retentionboards\_core')

app.config\_from\_object('django.conf:settings', namespace='CELERY')

app.autodiscover\_tasks(['retentionboards\_core.tasks'], force=True)

priority\_to\_routing\_key = {'high': 'hipri',

'mid': 'midpri',

'low': 'lopri'}

with app.pool.acquire(block=True) as conn:

exchange = Exchange(name='retention\_exchange', type='direct', durable=True, channel=conn)

queue\_task = Queue(name='retention\_queue\_task', exchange='retention\_exchange', routing\_key='task', message\_ttl=600,

durable=True, channel=conn)

app.conf.task\_queues=[queue\_task]

app.conf.task\_default\_exchange = 'retention\_exchange'

app.conf.task\_default\_exchange\_type = 'direct'

app.conf.task\_default\_routing\_key = 'task'

## **Файл retentionboards\_core/app/Dockerfile**

# pull official base image

FROM python:3.8.0-alpine

# set work directory

WORKDIR /usr/src/app

# set environment variables

ENV PYTHONDONTWRITEBYTECODE 1

ENV PYTHONUNBUFFERED 1

# install psycopg2 dependencies

RUN apk update \

&& apk add postgresql-dev gcc python3-dev musl-dev jpeg-dev zlib-dev

# install dependencies

RUN pip install --upgrade pip

COPY ./requirements.txt /usr/src/app/requirements.txt

RUN pip install --no-cache-dir -r requirements.txt

# copy entrypoint.sh

COPY ./entrypoint.sh /usr/src/app/entrypoint.sh

# copy project

COPY . /usr/src/app/

RUN chmod +x /usr/src/app/entrypoint.sh

# run entrypoint.sh

ENTRYPOINT ["/usr/src/app/entrypoint.sh"]

## **Файл retentionboards\_core/app/Dockerfile.prod**

###########

# BUILDER #

###########

# pull official base image

FROM python:3.7.0-alpine as builder

# set work directory

WORKDIR /usr/src/app

# set environment variables

ENV PYTHONDONTWRITEBYTECODE 1

ENV PYTHONUNBUFFERED 1

# install psycopg2 dependencies

RUN apk update \

&& apk add postgresql-dev gcc python3-dev musl-dev

# lint

#RUN pip install --upgrade pip

#RUN pip install flake8

#RUN flake8 --ignore=E501,F401 .

COPY . /usr/src/app/

# install dependencies

COPY ./requirements.txt .

RUN pip wheel --no-cache-dir --no-deps --wheel-dir /usr/src/app/wheels -r requirements.txt

#########

# FINAL #

#########

# pull official base image

FROM python:3.7.0-alpine

# create directory for the app user

RUN mkdir -p /home/app

# create the app user

RUN addgroup -S app && adduser -S app -G app

# create the appropriate directories

ENV HOME=/home/app

ENV APP\_HOME=/home/app/web

RUN mkdir $APP\_HOME

RUN mkdir $APP\_HOME/staticfiles

RUN mkdir $APP\_HOME/mediafiles

WORKDIR $APP\_HOME

# install dependencies

RUN apk update && apk add libpq

COPY --from=builder /usr/src/app/wheels /wheels

COPY --from=builder /usr/src/app/requirements.txt .

RUN pip install --upgrade pip

RUN pip install --no-cache /wheels/\*

# copy entrypoint-prod.sh

COPY ./entrypoint.prod.sh $APP\_HOME

# copy project

COPY . $APP\_HOME

# chown all the files to the app user

RUN chown -R app:app $APP\_HOME

# change to the app user

USER app

RUN chmod +x /home/app/web/entrypoint.prod.sh

# run entrypoint.prod.sh

ENTRYPOINT ["/home/app/web/entrypoint.prod.sh"]

## **Файл retentionboards\_core/app/entrypoint.sh**

#!/bin/sh

if [ "$DATABASE" = "postgres" ]

then

echo "Waiting for postgres..."

while ! nc -z $SQL\_HOST $SQL\_PORT; do

sleep 0.5

done

echo "PostgreSQL started"

fi

if [ "$WEB" = "TRUE" ]

then

python manage.py flush --no-input

python manage.py makemigrations

python manage.py migrate

python manage.py collectstatic --no-input --clear

python manage.py initadmin

fi

exec "$@"

## **Файл retentionboards\_core/app/entrypoint.prod.sh**

#!/bin/sh

if [ "$DATABASE" = "postgres" ]

then

echo "Waiting for postgres..."

while ! nc -z $SQL\_HOST $SQL\_PORT; do

sleep 0.5

done

echo "PostgreSQL started"

fi

exec "$@"

## **Файл retentionboards\_core/app/manage.py**

#!/usr/bin/env python

"""Django's command-line utility for administrative tasks."""

import os

import sys

def main():

os.environ.setdefault('DJANGO\_SETTINGS\_MODULE', 'retentionboards\_core.settings')

try:

from django.core.management import execute\_from\_command\_line

except ImportError as exc:

raise ImportError(

"Couldn't import Django. Are you sure it's installed and "

"available on your PYTHONPATH environment variable? Did you "

"forget to activate a virtual environment?"

) from exc

execute\_from\_command\_line(sys.argv)

if \_\_name\_\_ == '\_\_main\_\_':

main()

## **Файл retentionboards\_core/app/templates/base.html**

<!DOCTYPE HTML>

{% load static %}

<html>

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<title>RetentionBoards</title>

<meta name="viewport" content="width=device-width, initial-scale=1">

<meta name="description" content=""/>

<meta name="keywords" content=""/>

<meta name="author" content=""/>

<link href="https://fonts.googleapis.com/css?family=Poppins:300,400,500,600" rel="stylesheet">

<link href="https://fonts.googleapis.com/css?family=Nunito:200,300,400" rel="stylesheet">

<!-- For login form.css -->

<link rel="stylesheet" href="{% static 'css/newstyle.css' %}">

<!-- Animate.css -->

<link rel="stylesheet" href="{% static 'css/animate.css' %}">

<!-- Icomoon Icon Fonts-->

<link rel="stylesheet" href="{% static 'css/icomoon.css' %}">

<!-- Bootstrap -->

<link rel="stylesheet" href="{% static "css/bootstrap.css" %}">

<!-- Magnific Popup -->

<link rel="stylesheet" href="{% static "css/magnific-popup.css" %}">

<!-- Owl Carousel -->

<link rel="stylesheet" href="{% static "css/owl.carousel.min.css" %}">

<link rel="stylesheet" href="{% static "css/owl.theme.default.min.css" %}">

<!-- Theme style -->

<link rel="stylesheet" href="{% static "css/style.css" %}">

<!-- Modernizr JS -->

<script src="{% static 'js/modernizr-2.6.2.min.js' %}"></script>

<!-- FOR IE9 below -->

<!--[if lt IE 9]>

<script src="js/respond.min.js"></script>

<![endif]-->

</head>

<body>

<style>

[data-role="dynamic-fields"] > .form-inline + .form-inline {

}

[data-role="dynamic-fields"] > .form-inline [data-role="add"] {

display: none;

}

[data-role="dynamic-fields"] > .form-inline:last-child [data-role="add"] {

display: inline-block;

}

[data-role="dynamic-fields"] > .form-inline:last-child [data-role="remove"] {

display: none;

}

</style>

<div class="colorlib-loader"></div>

<div id="page">

<nav class="colorlib-nav" role="navigation">

<div class="top-menu">

<div class="container">

<div class="row">

<div class="col-md-2">

<div id="colorlib-logo"><a href="">RetentionBoards</a></div>

</div>

<div class="col-md-10 text-right menu-1">

<ul>

<li class="active"><a href="/web/app/">Home</a></li>

{% if user.is\_authenticated %}

<li><a href="/web/app/dashboard/">Dashboard</a></li>

{% endif %}

<!-- <li><a href="about.html">About</a></li>

<li><a href="contact.html">Contact</a></li> -->

{% if user.is\_authenticated %}

<li><a href="/web/app/logout/">Logout, {{ user.username }}</a></li>

{% else %}

<li>

<button onclick="document.getElementById('id01').style.display='block'"

style="width:auto;">LOGIN

</button>

</li>

<li>

<button onclick="document.getElementById('id03').style.display='block'"

style="width:auto;">REGISTER

</button>

</li>

{% endif %}

</ul>

</div>

</div>

</div>

</div>

</nav>

<section id="home" class="video-hero"

style="height: 100px; background-image: url({% static 'images/cover\_img\_1.jpg' %}); background-size:cover; background-position: center center;background-attachment:fixed;"

data-section="home">

<div class="overlay"></div>

<div class="display-t display-t2 text-center">

<div class="display-tc display-tc2">

<div class="container">

<div class="col-md-12 col-md-offset-0">

</div>

</div>

</div>

</div>

</section>

{% block content %}

{% endblock %}

<footer id="colorlib-footer">

<div class="copy">

<div class="container">

<div class="row">

<div class="col-md-12 text-center">

<p>

Copyright &copy;<script>document.write(new Date().getFullYear());</script>

All rights reserved | This template is made with love by <a

target="\_blank">AlexNerru</a>

<br>

Demo Images: <a href="http://unsplash.co/" target="\_blank">Unsplash</a>, <a

href="http://pexels.com/" target="\_blank">Pexels</a>

</p>

</div>

</div>

</div>

</div>

</footer>

</div>

{% if messages %}

<div id="id00" class="modal" style="display: block">

<div class="container" style="background-color:#f1f1f1">

<h4>Errors occured:</h4>

<ul class="messages">

{% for message in messages %}

<li{% if message.tags %} class="{{ message.tags }}"{% endif %}>{{ message }}</li>

{% endfor %}

</ul>

</div>

<div class="container" style="background-color:#f1f1f1">

<button type="button" onclick="document.getElementById('id00').style.display='none'"

class="cancelbtn">

Close

</button>

</div>

</div>

{% endif %}

<div class="gototop js-top">

<a href="#" class="js-gotop"><i class="icon-arrow-up2"></i></a>

</div>

<!-- jQuery -->

<script src="{% static 'js/jquery.min.js' %}"></script>

<!-- jQuery Easing -->

<script src="{% static 'js/jquery.easing.1.3.js' %}"></script>

<!-- Bootstrap -->

<script src="{% static 'js/bootstrap.min.js' %}"></script>

<!-- Waypoints -->

<script src="{% static 'js/jquery.waypoints.min.js' %}"></script>

<!-- Stellar Parallax -->

<script src="{% static 'js/jquery.stellar.min.js' %}"></script>

<!-- YTPlayer -->

<script src="{% static 'js/jquery.mb.YTPlayer.min.js' %}"></script>

<!-- Owl carousel -->

<script src="{% static 'js/owl.carousel.min.js' %}"></script>

<!-- Magnific Popup -->

<script src="{% static 'js/jquery.magnific-popup.min.js' %}"></script>

<script src="{% static 'js/magnific-popup-options.js' %}"></script>

<!-- Counters -->

<script src="{% static 'js/jquery.countTo.js' %}"></script>

<!-- Main -->

<script src="{% static 'js/main.js' %}"></script>

</body>

<script>

$(function () {

// Remove button click

$(document).on(

'click',

'[data-role="dynamic-fields"] > .form-inline [data-role="remove"]',

function (e) {

e.preventDefault();

$(this).closest('.form-inline').remove();

}

);

// Add button click

$(document).on(

'click',

'[data-role="dynamic-fields"] > .form-inline [data-role="add"]',

function (e) {

e.preventDefault();

var container = $(this).closest('[data-role="dynamic-fields"]');

new\_field\_group = container.children().filter('.form-inline:first-child').clone();

new\_field\_group.find('input').each(function () {

$(this).val('');

});

container.append(new\_field\_group);

}

);

});

</script>

<script>

// When the user clicks anywhere outside of the modal, close it

window.onclick = function (event) {

if (event.target.classList.contains('modal'))

event.target.style.display = "none";

}

</script>

<script>

var interval = 100; // 1000 = 1 second, 3000 = 3 seconds

function doAjax() {

var task = '{{task\_tsne}}';

var id = '{{eventset\_name}}';

$.ajax({

type: 'POST',

url: '/web/app/dashboard/check\_status/',

data: {'task\_id' : task},

csrfmiddlewaretoken: '{{ csrf\_token }}',

dataType: 'json',

success: function (data) {

$('#' + id).text(data);

},

complete: function (data) {

setTimeout(doAjax, interval);

}

});

}

doAjax();

</script>

</html>

## **Файл retentionboards\_core/app/templates/dashboard.html**

{% extends 'base.html' %}

{% block content %}

<div class="colorlib-blog">

<div class="container">

{% for row in eventsets|slice:"0:1" %}

<div class="row">

<div class="col-md-4 text-center animate-box"

onclick="document.getElementById('id02').style.display='block'">

<div class="services">

<article>

<a>

<span class="icon">

<i class="icon-plus"></i>

</span>

<h3>Click here to create set of events</h3>

</a>

</article>

</div>

</div>

{% for eventset in row %}

<div class="col-md-4 text-center animate-box">

<div class="services">

<article>

<a href={{ eventset.href }}>

<span class="icon">

<i class="icon-coin-dollar"></i>

</span>

<h2>{{ eventset.name }}</h2>

<h3 id={{eventset.name}}></h3>

</a>

</article>

</div>

</div>

{% endfor %}

</div>

{% endfor %}

{% for row in eventsets|slice:"1:" %}

<div class="row">

{% for eventset in row %}

<div class="col-md-4 text-center animate-box">

<div class="services">

<article>

<a href={{ eventset.href }}>

<span class="icon">

<i class="icon-coin-dollar"></i>

</span>

<h2>{{ eventset.name }}</h2>

<h3 id={{eventset.name}}></h3>

</a>

</article>

</div>

</div>

{% endfor %}

</div>

{% endfor %}

</div>

</div>

<div id="id02" class="modal">

<form class="modal-content animate" enctype="multipart/form-data" action="/web/app/dashboard/csv/" method="post">

<div style="padding-top: 1%; padding-left: 2%">

<h2>

Load csv file to import events data

</h2>

</div>

<div style="margin-left: 20px;">

{% csrf\_token %}

{{ form }}

<button type="submit" style="margin-top: 10px; padding-bottom: 10px">Upload</button>

</div>

</form>

</div>

{% endblock %}

## **Файл retentionboards\_core/app/templates/experiment.html**

{% extends 'base.html' %}

{% block content %}

{% load static %}

<style>

.x\_title .filter {

width: 40%;

float: right

}

.x\_title h2 {

margin: 5px 0 6px;

float: left;

display: block;

text-overflow: ellipsis;

overflow: hidden;

white-space: nowrap

}

.x\_title h2 small {

margin-left: 10px

}

.x\_title span {

color: #BDBDBD

}

.x\_panel {

position: relative;

margin-bottom: 10px;

padding: 10px 17px;

display: inline-block;

background: #fff;

border: 1px solid #E6E9ED;

-webkit-column-break-inside: avoid;

-moz-column-break-inside: avoid;

column-break-inside: avoid;

opacity: 1;

transition: all .2s ease;

}

.x\_title {

border-bottom: 2px solid #E6E9ED;

padding: 1px 5px 6px;

margin-bottom: 10px

}

.x\_title .filter {

width: 40%;

float: right

}

.x\_title h2 {

margin: 5px 0 6px;

float: left;

display: block;

text-overflow: ellipsis;

overflow: hidden;

white-space: nowrap

}

.x\_title h2 small {

margin-left: 10px

}

.x\_title span {

color: #BDBDBD

}

.x\_content {

padding: 0 5px 6px;

position: relative;

width: 100%;

float: left;

clear: both;

margin-top: 5px

}

.x\_content h4 {

font-size: 16px;

font-weight: 500

}

</style>

<div class="colorlib-blog">

<div class="container">

<div class="row">

<div class="col-md-12 animate-box">

<article>

<div class="text-right top-right">

<a href="{{ portfolio.delete\_href }}">

<i class="icon-cross"

style="color: red;%"></i>

</a>

</div>

<a>

<h3>Eventset name: {{ eventset.name }}</h3>

</a>

</article>

</div>

<div class="row">

<div class="col-md-3 text-center animate-box">

<div class="services">

<article>

<a>

<h3>Total users</h3>

<h2>{{ eventset.users }}</h2>

</a>

</article>

</div>

</div>

<div class="col-md-3 text-center animate-box">

<div class="services">

<article>

<a>

<h3>Total events</h3>

<h2>{{ eventset.events }}</h2>

</a>

</article>

</div>

</div>

<div class="col-md-3 text-center animate-box">

<div class="services">

<article>

<a>

<h3>Events per user</h3>

<h2>{{ eventset.events\_per\_user }}</h2>

</a>

</article>

</div>

</div>

<div class="col-md-3 text-center animate-box">

<div class="services">

<article>

<a>

<h3>Unique events</h3>

<h2>{{ eventset.unique\_events }}</h2>

</a>

</article>

</div>

</div>

</div>

<div class="row">

<div class="col-md-12 col-sm-12 col-xs-12 animate-box">

<div class="x\_panel">

<div class="x\_title">

<h2>Step Matrix</h2>

<div class="clearfix"></div>

</div>

<img src="{% static step\_matrix\_link %}" width="1100px" height="800px">

</div>

</div>

</div>

<div class="row">

<div class="col-md-12 col-sm-12 col-xs-12 animate-box">

<div class="x\_panel">

<div class="x\_title">

<h2>TSNE</h2>

<div class="clearfix"></div>

</div>

<img src="{% static tsne\_link %}" width="1100px" height="800px" style="align-self: center">

</div>

</div>

</div>

{% for graph\_element in graphs\_all %}

<div class="row">

<div class="col-md-12 col-sm-12 col-xs-12 text-center">

<article>

<a href={{ graph\_element.0 }}>

<span class="icon">

<i class="icon-bar-graph"></i>

</span>

<h2>{{ graph\_element.1 }}</h2>

</a>

</article>

</div>

</div>

{% endfor %}

</div>

</div>

</div>

{% endblock %}

## **Файл retentionboards\_core/app/templates/graph.html**

{% extends 'base.html' %}

{% block content %}

{% load static %}

<style>

.x\_title .filter {

width: 40%;

float: right

}

.x\_title h2 {

margin: 5px 0 6px;

float: left;

display: block;

text-overflow: ellipsis;

overflow: hidden;

white-space: nowrap

}

.x\_title h2 small {

margin-left: 10px

}

.x\_title span {

color: #BDBDBD

}

.x\_panel {

position: relative;

margin-bottom: 10px;

padding: 10px 17px;

display: inline-block;

background: #fff;

border: 1px solid #E6E9ED;

-webkit-column-break-inside: avoid;

-moz-column-break-inside: avoid;

column-break-inside: avoid;

opacity: 1;

transition: all .2s ease;

width: 1300px;

margin-left: 0px;

}

.x\_title {

border-bottom: 2px solid #E6E9ED;

padding: 1px 5px 6px;

margin-bottom: 10px

}

.x\_title .filter {

width: 40%;

float: right

}

.x\_title h2 {

margin: 5px 0 6px;

float: left;

display: block;

text-overflow: ellipsis;

overflow: hidden;

white-space: nowrap

}

.x\_title h2 small {

margin-left: 10px

}

.x\_title span {

color: #BDBDBD

}

.x\_content {

padding: 0 5px 6px;

position: relative;

width: 100%;

float: left;

clear: both;

margin-top: 5px

}

.x\_content h4 {

font-size: 16px;

font-weight: 500

}

</style>

<div class="colorlib-blog">

<div class="container">

<div class="row">

<div class="row">

<div class="col-md-12 col-sm-12 col-xs-12 animate-box">

<div class="x\_panel" style="margin-left: 0">

<div class="x\_title">

<div class="clearfix"></div>

</div>

<div class="x\_content">

{% if graph %}

<div>

{{ graph|safe }}

</div>

{% endif %}

</div>

</div>

</div>

</div>

</div>

</div>

</div>

{% endblock %}

## **Файл retentionboards\_core/app/templates/index.html**

{% extends 'base.html' %}

{% block content %}

<div class="colorlib-services">

<div class="container">

<div class="row">

<div class="col-md-4 text-center animate-box">

<div class="services">

<span class="icon">

<i class="icon-search"></i>

</span>

<div class="desc">

<h3>Analyze users behavior</h3>

<p>Understand why customers leave your product and reduce churn rate</p>

</div>

</div>

</div>

<div class="col-md-4 text-center animate-box">

<div class="services">

<span class="icon">

<i class="icon-road"></i>

</span>

<div class="desc">

<h3>View trajectories</h3>

<p>Explore how customers use your app or website</p>

</div>

</div>

</div>

<div class="col-md-4 text-center animate-box">

<div class="services">

<span class="icon">

<i class="icon-coin-dollar"></i>

</span>

<div class="desc">

<h3>Visualize business</h3>

<p>Visualize the analytics results to see business growth points</p>

</div>

</div>

</div>

</div>

</div>

</div>

<div id="id01" class="modal">

<form class="modal-content animate" action="/web/app/login/" method="post">

<div class="container">

{% csrf\_token %}

{{ form }}

<div class="row" style="margin-top:10px">

<button class="col-md-5 btn-success" type="submit">Login</button>

<div class="col-md-2"> </div>

<button type="button" onclick="document.getElementById('id01').style.display='none'"

class="btn-danger cancelbtn col-md-5">

Cancel

</button>

</div>

</div>

</form>

</div>

<div id="id03" class="modal">

<form class="modal-content animate" action="/web/app/register/" method="post">

<div class="container">

{% csrf\_token %}

{{ form\_register }}

<div class="row" style="margin-top:10px;">

<button type="submit" class="col-md-5 btn-success" style="margin-left: 10px">Register</button>

<button type="button" class="btn-danger cancelbtn col-md-5" style="margin-left: 120px" onclick="document.getElementById('id03').style.display='none'">

Cancel

</button>

</div>

</div>

</form>

</div>

{% endblock %}

## **Файл retentionboards\_core/nginx/Dockerfile**

FROM nginx:1.17.0-alpine

RUN rm /etc/nginx/conf.d/default.conf

COPY nginx.conf /etc/nginx/conf.d

## **Файл retentionboards\_core/nginx/nginx.conf**

upstream retentionboards {

server web:8000;

}

server {

listen 80;

ssl off;

location / {

proxy\_pass http://retentionboards;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header Host $host;

proxy\_redirect off;

}

location /staticfiles/ {

alias /usr/src/app/staticfiles/;

}

location /mediafiles/ {

alias /usr/src/app/mediafiles/;

}

}

## **Файл docker-compose.prod.yml**

version: '3.7'

services:

web:

build: retentionboards\_core/app

command: gunicorn retentionboards\_core.wsgi:application --bind 0.0.0.0:8000

volumes:

- static\_volume:/home/app/web/staticfiles

- media\_volume:/home/app/web/mediafiles

expose:

- 8000

env\_file:

- env.prod.web

depends\_on:

- db

retentioneering:

build: retentionboards\_retentioneering/app

command: gunicorn retentionboards\_retentioneering.wsgi:application --bind 0.0.0.0:8001

volumes:

- retentioneering:/usr/src/app/

env\_file:

- env.prod.services

depends\_on:

- db

- redis\_queue

- redis\_cache

- web

deploy:

replicas: 4

db:

image: postgres:12.0-alpine

volumes:

- postgres\_data:/var/lib/postgresql/data/

env\_file:

- env.prod.db

redis\_queue:

image: redis:latest

volumes:

- redis\_queue\_data:/data

celery:

build: retentionboards\_core/app

command: sh -c "celery -A retentionboards\_core worker -l info"

volumes:

- celery\_data:/code

env\_file:

- env.prod.services

depends\_on:

- db

- redis\_queue

- web

celery\_beat:

build: retentionboards\_core/app

command: sh -c "celery -A retentionboards\_core beat -l debug"

volumes:

- celery\_beat:/code

env\_file:

- env.prod.services

depends\_on:

- db

- redis\_queue

- web

celery\_retentioneering:

build: retentionboards\_retentioneering/app

command: sh -c "celery -A retentionboards\_retentioneering worker -l info --concurrency=10"

volumes:

- celery\_rete:/code

env\_file:

- env.prod.web

depends\_on:

- db

- redis\_queue

- retentioneering

- web

flower:

image: mher/flower

command: ["flower", "--broker=redis://redis\_queue:6379/0", "--port=8888"]

ports:

- 8888:8888

depends\_on:

- celery

- celery\_beat

- celery\_retentioneering

- redis\_queue

redis\_cache:

image: redis:latest

volumes:

- redis\_cache\_data:/data

volumes:

postgres\_data:

redis\_queue\_data:

redis\_cache\_data:

celery\_data:

app:

retentioneering:

celery\_beat:

celery\_rete:

## **Файл docker-compose.yml**

version: '3.7'

services:

web:

build: retentionboards\_core/app

command: sh -c "python manage.py runserver 0.0.0.0:8000"

volumes:

- app:/usr/src/app/

ports:

- 8000:8000

env\_file:

- env.dev.web

depends\_on:

- db

- redis\_queue

- redis\_cache

retentioneering:

build: retentionboards\_retentioneering/app

command: sh -c "python manage.py runserver 0.0.0.0:8001"

volumes:

- retentioneering:/usr/src/app/

env\_file:

- env.dev.services

depends\_on:

- db

- redis\_queue

- redis\_cache

- web

deploy:

replicas: 4

db:

image: postgres:12.0-alpine

volumes:

- postgres\_data:/var/lib/postgresql/data/

env\_file:

- env.dev.db

ports:

- 5432:5432

redis\_queue:

image: redis:latest

volumes:

- redis\_queue\_data:/data

celery:

build: retentionboards\_core/app

command: sh -c "celery -A retentionboards\_core worker -l info"

volumes:

- celery\_data:/code

env\_file:

- env.dev.services

depends\_on:

- db

- redis\_queue

- web

celery\_beat:

build: retentionboards\_core/app

command: sh -c "celery -A retentionboards\_core beat -l debug"

volumes:

- celery\_beat:/code

env\_file:

- env.dev.services

depends\_on:

- db

- redis\_queue

- web

celery\_retentioneering:

build: retentionboards\_retentioneering/app

command: sh -c "celery -A retentionboards\_retentioneering worker -l info --concurrency=10"

volumes:

- celery\_rete:/code

env\_file:

- env.dev.web

depends\_on:

- db

- redis\_queue

- retentioneering

- web

flower:

image: mher/flower

command: ["flower", "--broker=redis://redis\_queue:6379/0", "--port=8888"]

ports:

- 8888:8888

depends\_on:

- celery

- celery\_beat

- celery\_retentioneering

- redis\_queue

redis\_cache:

image: redis:latest

volumes:

- redis\_cache\_data:/data

volumes:

postgres\_data:

redis\_queue\_data:

redis\_cache\_data:

celery\_data:

app:

retentioneering:

celery\_beat:

celery\_rete:

## **Файл retentionboards\_retentioneering/app/analytics/tasks.py**

from \_\_future\_\_ import absolute\_import, unicode\_literals

import pandas as pd

from retentioneering\_library.visualization.plot import step\_matrix, cluster\_tsne

from retentioneering\_library.core.utils import init\_config

from time import sleep

import psycopg2

from psycopg2.extras import RealDictCursor

import os

from celery import shared\_task, current\_task

from celery import states

from .redis\_connector import save\_image\_to\_redis, save\_html\_to\_redis

query\_string = """

select

event\_name as event\_name,

event\_timestamp as event\_timestamp,

user\_pseudo\_id as user\_pseudo\_id

from dashboards\_event

where event\_set\_id = {}

order by event\_timestamp"""

@shared\_task(name='prepare\_dataset')

def prepare\_dataset(eventset\_id):

current\_task.update\_state(state=states.STARTED)

conn = psycopg2.connect(

database='hello\_django\_dev',

host='db',

port='5432',

user=os.environ.get("SQL\_USER", "user"),

password=os.environ.get("SQL\_PASSWORD", "password"),

cursor\_factory=RealDictCursor

)

init\_config(

experiments\_folder='experiments',

index\_col='user\_pseudo\_id', # column by which we split users / sessions / whatever

event\_col='event\_name', # column that describes event

event\_time\_col='event\_timestamp', # column that describes timestamp of event

positive\_target\_event='lost', # name of positive target event

negative\_target\_event='passed', # name of negative target event

pos\_target\_definition={

# how to define positive event, e.g. empty means that add passed for whom was not 'lost'

'time\_limit': 600

},

neg\_target\_definition={ # how to define negative event, e.g. users who were inactive for 600 seconds.

},

)

data = pd.read\_sql\_query(query\_string.format(eventset\_id), con=conn)

data = data.sort\_values('event\_timestamp')

desc\_table = data.retention.get\_step\_matrix(max\_steps=30, plot\_type=False)

heatmap = step\_matrix(

desc\_table.round(2),

title='Step matrix')

save\_image\_to\_redis(eventset\_id, 'heatmap', heatmap[0])

return eventset\_id

@shared\_task(name='learn\_tsne')

def learn\_tsne(eventset\_id):

current\_task.update\_state(state=states.STARTED)

conn = psycopg2.connect(

database='hello\_django\_dev',

host='db',

port='5432',

user=os.environ.get("SQL\_USER", "user"),

password=os.environ.get("SQL\_PASSWORD", "password"),

cursor\_factory=RealDictCursor

)

init\_config(

experiments\_folder='experiments',

index\_col='user\_pseudo\_id', # column by which we split users / sessions / whatever

event\_col='event\_name', # column that describes event

event\_time\_col='event\_timestamp', # column that describes timestamp of event

positive\_target\_event='lost', # name of positive target event

negative\_target\_event='passed', # name of negative target event

pos\_target\_definition={

# how to define positive event, e.g. empty means that add passed for whom was not 'lost'

'time\_limit': 600

},

neg\_target\_definition={ # how to define negative event, e.g. users who were inactive for 600 seconds.

},

)

data = pd.read\_sql\_query(query\_string.format(eventset\_id), con=conn)

data = data.sort\_values('event\_timestamp')

n\_clusters = 8

clusters, graph = data.retention.get\_clusters(perplexity=10, n\_clusters=n\_clusters,

plot\_type='cluster\_tsne', refit\_cluster=True)

save\_image\_to\_redis(eventset\_id, 'tsne', graph[0])

graphs = []

for cluster in range(n\_clusters):

try:

result = data.retention.filter\_cluster(cluster).retention.plot\_graph(width=1400, height=1000)

redis\_id = save\_html\_to\_redis(eventset=eventset\_id, key='cluster', name=result, cluster=cluster)

graphs.append(redis\_id)

except:

print("error")

return [eventset\_id, graphs]

## **Файл retentionboards\_retentioneering/app/analytics/redis\_connector.py**

from redis import Redis

import os

import pickle

from PIL import Image

redis = Redis(host=os.environ.get("REDIS\_HOST", 'redis'), port=os.environ.get("REDIS\_PORT", 6379))

def save\_image\_to\_redis(eventset, key, image):

directory = f'experiments/{eventset}/'

filename = f'experiments/{eventset}/{key}.png'

if not os.path.exists(directory):

os.makedirs(directory)

image.figure.savefig(filename)

im = Image.open(filename)

data = pickle.dumps(im)

redis\_id = key + str(eventset)

redis.set(redis\_id, data)

os.remove(filename)

return redis\_id

def save\_html\_to\_redis(eventset, key, name, cluster):

with open(name, 'r') as f:

html = f.read()

data = pickle.dumps(html)

redis\_id = key + str(eventset) + '\_' + str(cluster)

redis.set(redis\_id, data)

return redis\_id

# **ПРИЛОЖЕНИЕ 1**

# **ТЕРМИНОЛОГИЯ**

|  |  |
| --- | --- |
| **Термин** | **Определение** |
| **Система аналитики** | Система, которая позволяет собирать, хранить и обрабатывать события аналитики |
| **Событие аналитик** | Действие пользователя в приложении, характеризуемое названием, пользователем и временем |
| **Матрица переходов** | Матрица вероятности следующего события в зависимости от предыдущего |
| **Граф перемещений** | Визуализация матрицы перехода в виде графа |
| **Поведение пользователя** | Порядок действий осуществленным пользователем в программном продукте выраженная как последовательность событий аналитики |

# **ПРИЛОЖЕНИЕ 2**

# **СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ**

1. ГОСТ 19.101-77 Виды программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
2. ГОСТ 19.102-77 Стадии разработки. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
3. ГОСТ 19.103-77 Обозначения программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
4. ГОСТ 19.104-78 Основные надписи. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
5. ГОСТ 19.105-78 Общие требования к программным документам. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
6. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
7. ГОСТ 19.603-78 Общие правила внесения изменений. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
8. ГОСТ 19.604-78 Правила внесения изменений в программные документы, выполненные печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.

# **ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Изм. | Номера листов (страниц) | | | | Всего листов (страниц) в документе | № документа | Входящий № сопроводительного документа и дата | Подпись | Дата |
| измененных | замененных | новых | аннулированных |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |