Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего профессионального образования

‹‹Уральский федеральный университет имени первого Президента России

Б.Н.Ельцина››

Кафедра ‹‹Интеллектуальных информационных технологий››

дисциплина ‹‹WEB программирование››

Курсовая работа

«WebNews» - новостной мультимедийный портал

Преподаватель: \_\_\_\_\_\_\_\_\_\_\_\_\_ Свинцов Д. В.

(подпись преподавателя)

Студент: \_\_\_\_\_\_\_\_\_\_\_\_\_ Сарафанов Д.Е.

(подпись студента)

Екатеринбург - 2016

Оглавление

[Введение 3](#_Toc435820379)

[Описание проекта 3](#_Toc435820380)

[Обоснование проекта 3](#_Toc435820381)

[Инструкция пользователя 4](#_Toc435820382)

[Главная страница 4](#_Toc435820383)

[Страница авторизации 5](#_Toc435820384)

[Страница «Игра» 5](#_Toc435820385)

[Резактирование базы данных (Админка) 5](#_Toc435820386)

[Инструкция программиста 6](#_Toc435820387)

[Установка 6](#_Toc435820388)

[Инициализация баз данных 6](#_Toc435820389)

[Запуск проекта 6](#_Toc435820390)

[Авторизация 6](#_Toc435820391)

[Описание функций, классов и блоков кода 7](#_Toc435820392)

[Выводы 14](#_Toc435820393)

# Введение

Необходимо создать веб-сервер и сайт к нему в рамках курса web-программирования. Необходимо было использовать фреймворк pyramid для написания веб-сервера. Выбор темы определяется лично каждым студентом.

В качестве темы для сайта был выбран новостной портал, так как такой сайт подходит по требованиям к курсовой работе.

# Описание сайта

Сайт состоит из следующих страниц:

* Главная — содержит список всех новостей;
* Новость — содержит подробную информацию по выбранной новости;
* Авторизация — содержит форму авторизации пользователей;
* Редактирование — содержит форму создания новости.

# Инструкция пользователя

Сайт состоит из 4 страниц, переход на каждую из которых, производится нажатием на соответствующий элемент страницы.

## Главная страница

Данная страница отображается когда пользователь заходит на сайт. На данной странице отображаются следующие элементы:

1. Список всех новостей;
2. Логотип сайта;
3. Общие, для всех страниц, элементы управления перехода на главную страницу и переход на страницу авторизации.

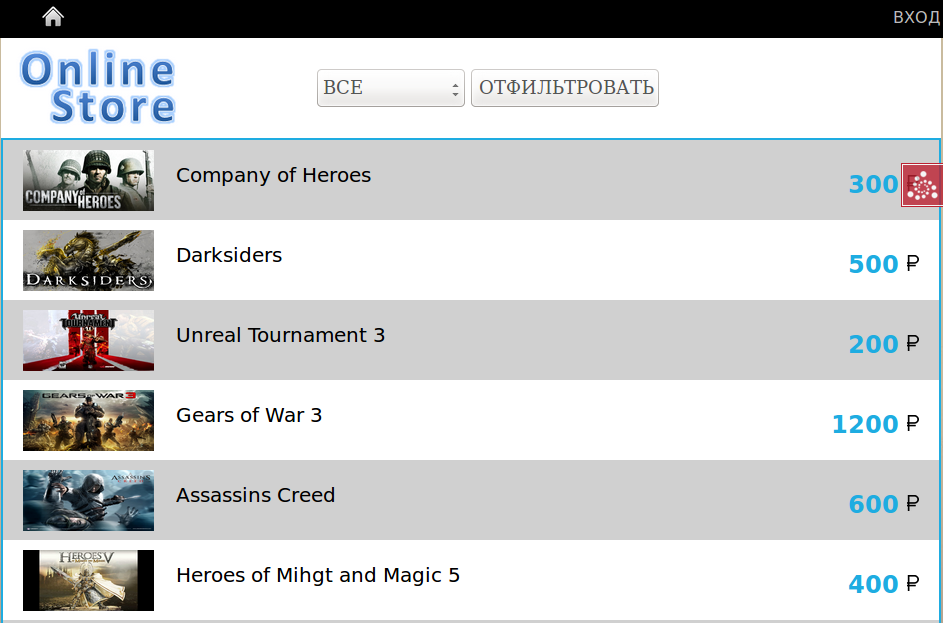


Рис.1. Главная страница

## Страница авторизации

Открывается при нажатии на ссылку «вход» на главной странице. Содержит следующие элементы :

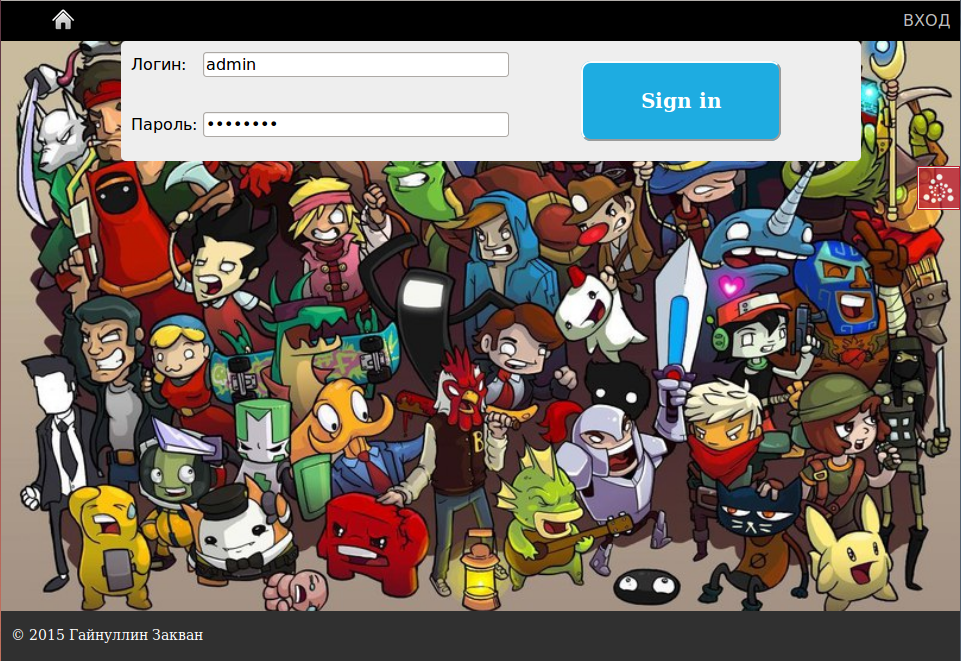
* Поле ввода логина
* Поле ввода пароля
* Кнопка подтверждения логина и пароля

Рис.2. Страница авторизации

## Страница «Игра»

Эта страница открывается при нажатии на любой из элементов списка игра на главной странице. На ней показана информация о игре и ее изображение. При нажатии на кнопку «Капить» на экране появляется код активации игры.

## Резактирование базы данных (Админка)

Эта страница доступна по пути localhost:6543/admin/ или при клике на логин авторизовавшегося пользователя в правом верхнем углу экрана.

# Инструкция программиста

## Установка

Для установки требуется версия Python 2.7, наличие установленных пакетов Jinja2 и PyramidFramework.

В командной строке необходимо создать виртуальной пространство и перейти в папку проекта. Далее пишем команду:

* Python setup.py develop

## Инициализация баз данных

Для инициализации баз данных прописываем в командной строке следующую команду:

* Initialize\_OnlineStore\_db decelopment.ini

## Запуск проекта

Запустить проект можно следующей командой:

* pserve development.ini

## Авторизация

Для того, чтобы зайти на сайт в качестве администратора, нужно перейти на страницу с формой авторизации, нажав на ссылку в правом верхнем углу.

Логин и пароль указаны в файле security.py.  
Если пользователь авторизован, логин указывается в правом верхнем углу страниц сайта.

# Описание функций, классов и блоков кода

* В файл \_\_init.py\_\_ я добавил роуты для страниц сайта и настройки для авторизации и аутентификации.

from pyramid.config import Configurator

from sqlalchemy import engine\_from\_config

from pyramid.authentication import AuthTktAuthenticationPolicy

from pyramid.authorization import ACLAuthorizationPolicy

from .security import groupfinder

from .models import (

DBSession,

Base,

Game,

Genre,

GameDeveloper,

Publisher)

def main(global\_config, \*\*settings):

""" This function returns a Pyramid WSGI application. """

engine = engine\_from\_config(settings, 'sqlalchemy.')

DBSession.configure(bind=engine)

Base.metadata.bind = engine

authentific\_policy = AuthTktAuthenticationPolicy( 'sosecret', callback=groupfinder, hashalg='sha512')

authoriz\_policy = ACLAuthorizationPolicy()

config = Configurator(settings=settings)

config.set\_authentication\_policy(authentific\_policy)

config.set\_authorization\_policy(authoriz\_policy)

config.include('pyramid\_chameleon')

config.include('pyramid\_sqlalchemy')

config.include('pyramid\_jinja2')

config.include('pyramid\_sacrud', route\_prefix='admin')

config.add\_static\_view('static', 'static')

config.add\_route('page', '/')

config.add\_route('login', '/login')

config.add\_route('logout', '/logout')

config.add\_route('game', 'game/{id}')

settings['pyramid\_sacrud.models'] = (('Project', [Game, Genre, GameDeveloper, Publisher]))

config.scan()

return config.make\_wsgi\_app()

* Файл views.py содержит представления для страниц сайта.

from pyramid.response import Response

from pyramid.view import view\_config

from sqlalchemy.exc import DBAPIError

from .models import(DBSession, Game, Genre, GameDeveloper, Publisher)

from pyramid.httpexceptions import(HTTPFound, HTTPNotFound)

from pyramid.view import(view\_config, forbidden\_view\_config)

from pyramid.security import(remember, forget)

from .security import USERS

@view\_config(route\_name='game', renderer='templates/game.jinja2')

def game\_view(request):

game\_id = request.matchdict['id']

game = DBSession.query(Game).filter\_by(id=game\_id).first()

return {'game': game,

'username': request.authenticated\_userid

}

@view\_config(route\_name='page', renderer='templates/page.jinja2')

def page\_view(request):

genres = DBSession.query(Genre)

games = DBSession.query(Game)

genre = None

if 'genre\_filter' in request.GET:

genre\_id = request.GET['genre\_filter']

genre = genres.filter\_by(id=genre\_id).first()

if genre:

games = games.filter\_by(genre=genre).all()

return {'games': games,

'genres': genres,

'selected\_genre': genre,

'username': request.authenticated\_userid

}

conn\_err\_msg = """\

Pyramid is having a problem using your SQL database. The problem

might be caused by one of the following things:

1. You may need to run the "initialize\_OnlineStore\_db" script

to initialize your database tables. Check your virtual

environment's "bin" directory for this script and try to run it.

2. Your database server may not be running. Check that the

database server referred to by the "sqlalchemy.url" setting in

your "development.ini" file is running.

After you fix the problem, please restart the Pyramid application to

try it again. """

@view\_config(route\_name='login', renderer='templates/login.jinja2')

@forbidden\_view\_config(renderer='templates/login.jinja2')

def login(request):

if 'submitted' in request.params:

login = request.params['login']

password = request.params['password']

if USERS.get(login) == password:

headers = remember(request, login)

return HTTPFound(location = '/', headers = headers)

else:

return {'message': 'Incorrect login or password', 'project': 'OnlineStore'};

return {'project': 'OnlineStore'}

@view\_config(route\_name='logout')

def logout(request):

headers = forget(request)

return HTTPFound(location = request.route\_url('page'), headers = headers)

* В файл models.py были описаны таблицы базы данных и права для групп пользователей.

from sqlalchemy import (Column, Index, Integer, Text, Boolean, ForeignKey)

from sqlalchemy.ext.declarative import declarative\_base

from sqlalchemy.orm import (scoped\_session, sessionmaker, relationship)

from pyramid.security import (Allow, Everyone)

from zope.sqlalchemy import ZopeTransactionExtension

DBSession = scoped\_session(sessionmaker(extension=ZopeTransactionExtension()))

Base = declarative\_base()

class Genre(Base):

\_\_tablename\_\_ = 'genres'

id = Column(Integer, primary\_key=True)

name = Column(Text, primary\_key=True)

info = Column(Text)

def \_\_repr\_\_(self):

return self.name or ''

class GameDeveloper(Base):

\_\_tablename\_\_ = 'game developers'

id = Column(Integer, primary\_key=True)

company\_name = Column(Text, primary\_key=True)

year\_of\_creation = Column(Integer)

country = Column(Text)

city = Column(Text)

def \_\_repr\_\_(self):

return self.company\_name + ' ' + self.country

class Publisher(Base):

\_\_tablename\_\_ = 'publishers'

id = Column(Integer, primary\_key=True)

company\_name = Column(Text, primary\_key=True)

year\_of\_creation = Column(Integer)

country = Column(Text)

city = Column(Text)

site = Column(Text, primary\_key=True)

def \_\_repr\_\_(self):

return self.company\_name + ' ' + self.country + ' ' + self.city

class Game(Base):

\_\_tablename\_\_ = 'games'

id = Column(Integer, primary\_key=True)

name = Column(Text, primary\_key=True)

price = Column(Integer)

image\_name = Column(Text)

genre\_id = Column(Integer, ForeignKey('genres.id'))

genre = relationship(Genre)

developer\_id = Column(Integer, ForeignKey('game developers.id'))

developer = relationship(GameDeveloper)

publisher\_id = Column(Integer, ForeignKey('publishers.id'))

publisher = relationship(Publisher)

def \_\_repr\_\_(self):

return self.name + ' ' + self.price + ' ' + self.genre

class Accesses(object):

\_\_name\_\_ = None

\_\_acl\_\_ = [ (Allow, 'group:editors', 'edit'),

(Allow, Everyone, 'view')]

def \_\_init\_\_(self, request):

pass

* Файл secutiry.py содержит группы пользователей.

USERS = {'admin':'password','admin2':'123'}

GROUPS = {'admin':['groups:editors'], 'admin2':['groups:editors']}

def groupfinder(userid, request):

if userid in USERS:

return GROUPS.get(userid, [])

# Выводы

Во время разработки сайта я использовал следующие технологии:

* Фреймворк Pyramid
* Шаблонизатор Jinja2
* ОРМ SQLAlchemy
* SQLite

В процессе работы я создал сайт при помощи фреймворка Pyramid с использованием шаблонов, а также подключать к проекту базу данных SQLite и работать с ней. А также научился создавать группы пользователей и авторизовать пользователей на сайте.