МИНИСТЕРСТВО науки и высшего ОБРАЗОВАНИЯ РОссИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ

(национальный исследовательский университет)»

Кафедра 319 «Интеллектуальные системы мониторинга»

**КУРСОВАЯ РАБОТА**

по дисциплине

«Основы построения промышленных программных систем»

**«Проектирование и разработка веб-приложения классификации текстов с применением методов машинного обучения»**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | | |  | Родионов В. С. |
| Группа | | | М3О-221М-20 |  |
| Руководитель | | |  | Полицына Е. В. |
| Оценка |  | Дата защиты «25» декабря 2021 г. | | |

**Москва 2021**

МИНИСТЕРСТВО науки и высшего ОБРАЗОВАНИЯ РОссИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ

(национальный исследовательский университет)»

Кафедра 319 «Интеллектуальные системы мониторинга»

УТВЕРЖДАЮ

Заведующий кафедрой

д.т.н., профессор Нагибин С.Я.

(И.О.Фамилия)

«\_\_\_\_»\_\_\_\_\_\_\_ 2021 г.

**З А Д А Н И Е**

на курсовую работу по дисциплине

Основы построения промышленных программных систем

Студент М3О-221М-20, Родионов Вадим Станиславович

(№ группы, Ф. И. О.)

Тема «Проектирование и разработка веб-приложения классификации текстов с применением методов машинного обучения»

Перечень вопросов, подлежащих разработке в курсовой работе

Реализовать клиент-серверную систему классификации для свой предметной области, которая представляет собой веб-приложение на Java/Python с пользовательским веб- интерфейсом.

Рекомендуемая литература

1 Gundecha U., Selenium Testing Tools Cookbook

2 Чакон С., Страуб Б., Pro Git

3 Лутц М., Изучаем Python

Задание выдано «12» сентября 2021 г.

Руководитель Полицына Е.В., к.т.н. доцент кафедры 319\_\_\_\_\_\_

(Ф. И. О., должность, подпись)

Студент\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

**СОДЕРЖАНИЕ**

[1 ОПИСАНИЕ ВОЗМОЖНОСТЕЙ ПРИЛОЖЕНИЯ 4](#_Toc89383112)

[1.1 Общие сведения 4](#_Toc89383113)

[*1.1.1 Полное наименование системы и её условное обозначение 4*](#_Toc89383114)

[*1.1.2 Наименование предприятий (объединений) разработчика и заказчика (пользователя) системы и их реквизиты 4*](#_Toc89383115)

[*1.1.3 Основания для разработки 4*](#_Toc89383116)

[*1.1.4 Плановые сроки начала и окончания работы по созданию системы 4*](#_Toc89383117)

[*1.1.5 Порядок оформления и предъявления заказчику результатов работ по созданию системы (её частей), по изготовлению и наладке отдельных средств (технических, программных, информационных) и программно-технических (программно-методических) комплексов истемы. 4*](#_Toc89383118)

[1.2 Назначения и цели 5](#_Toc89383119)

[*1.2.1 Назначение системы 5*](#_Toc89383120)

[*1.2.2 Цели создания системы 5*](#_Toc89383121)

[2 ТРЕБОВАНИЯ 6](#_Toc89383122)

[2.1 Требования к системе в целом 6](#_Toc89383123)

[2.2 Функциональные требования 6](#_Toc89383124)

[2.3 Нефункциональные требования 7](#_Toc89383125)

[3 АРХИТЕКТУРА ПРИЛОЖЕНИЯ 8](#_Toc89383126)

[4 ОПИСАНИЕ КЛАССИФИКАТОРА 9](#_Toc89383127)

[4.1 Классы 9](#_Toc89383128)

[4.2 Вектор признаков 9](#_Toc89383129)

[4.3 Обучающие и тестовые коллекции 9](#_Toc89383130)

[5 ОПИСАНИЕ ИНФРАСТРУКТУРЫ РАЗРАБОТКИ 10](#_Toc89383131)

[6 РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ 11](#_Toc89383132)

[6.1 Сценарии тестирования 11](#_Toc89383133)

[*6.1.2 Автоматическое тестирование клиентской части. 11*](#_Toc89383134)

[*6.1.1 Сценарий ручного тестирования клиентской части 11*](#_Toc89383135)

[6.2 Результаты тестирования 11](#_Toc89383136)

[7 ИНТЕРФЕЙС И ВОЗМОЖНОСТИ СИСТЕМЫ 12](#_Toc89383137)

[8 АНАЛИЗ РЕЗУЛЬТАТОВ 16](#_Toc89383138)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 17](#_Toc89383139)

# 1 ОПИСАНИЕ ВОЗМОЖНОСТЕЙ ПРИЛОЖЕНИЯ

## 1.1 Общие сведения

### 1.1.1 Полное наименование системы и её условное обозначение

Веб-приложение «Django Unchained для коллекций статей с сайта newsru.com. Просмотр, классификация, редактирование».

*Условное обозначение:* Классификатор категорий по текстам.

### 1.1.2 Наименование предприятий (объединений) разработчика и заказчика (пользователя) системы и их реквизиты

Заказчик – кафедра 319, МАИ (национальный исследовательский университет), Полицына Е. В.

Разработчик – студент группы М3О-221М-20 Родионов В. С.

### 1.1.3 Основания для разработки

Курсовая работа по предмету «Основы построения промышленных программных систем», целью которой является изучение методов машинного обучения, проектирование, моделирование, разработка и тестирование веб-приложения с использованием изученных методов классификации.

### 1.1.4 Плановые сроки начала и окончания работы по созданию системы

Начало работ по проектированию и реализации системы: сентябрь 2021 г.

Окончание работ по созданию системы: декабрь 2021 г

### 1.1.5 Порядок оформления и предъявления заказчику результатов работ по созданию системы (её частей), по изготовлению и наладке отдельных средств (технических, программных, информационных) и программно-технических (программно-методических) комплексов системы.

К результатам труда разработчика относится:

* оригинальное программное обеспечение;
* оригинальные модели классификаторов;
* оригинальные алгоритмы работы программы;
* проектная и рабочая документация.

Заказчику передаются:

* ссылка на репозиторий с исходным кодом проекта;
* исполняемый файл проекта и необходимые модели классификатора;
* устная просьба поставить оценку «Отлично»;
* сопроводительная документация проекта.

Проектная документация должна быть разработана в соответствии с ГОСТ 34.602-89. Процедуры приемки – передачи результатов работ оформляются актами приемки-передачи.

## 1.2 Назначения и цели

### 1.2.1 Назначение системы

Классификатор категорий из введённого текста предназначен для автоматизации процесса классифицирования объектов по 10 следующим классам:

* Автоновости;
* В мире;
* В России;
* Инопресса;
* Культура;
* Медицина;
* Недвижимость;
* Спорт;
* Технологии;
* Экономика;

А также из текста, введённым пользователем.

### 1.2.2 Цели создания системы

Целью создания системы является:

* автоматизация процесса хранения текстов;
* автоматизация классификации текста;
* анализ результатов практического применения машинного обучения.

# 2 ТРЕБОВАНИЯ

## 2.1 Требования к системе в соответствии с ТЗ

Система должна позволять:

1. Загрузить текст и как минимум ответить на вопрос, соответствует ли загруженный текст предметной области системы (бинарная классификация на основе методов машинного обучения);

2. При построении модели классификатора в качестве способа векторизации можно использовать любой, минимум “bag-of-words”;

3. Предоставлять возможность пользователю через веб-интерфейс просмотреть объем коллекции и примеры документов в системе с возможностью задания фильтров по каким-либо признакам, которые есть в корпусе текстов системы (например, “автор”).

Требования к реализации:

1. Бэкенд должен быть реализован на Java/Python;

2. Оформление кода должно соответствовать Java Style Guide/PEP8;

3. Для реализации фронтенда могут быть использованы любые технологии.

4. Для реализации классификатора могут быть использованы любые библиотеки, реализующие методы машинного обучения;

5. Для реализации векторизации документов или других функций могут быть использованы любые библиотеки, фреймворки, внешние API и т.д.;

6. Обязательно использование системы контроля версий для хранения всех артефактов проекта (предпочтительно GitHub);

7. Обязательно поддержание проекта в работоспособном состоянии с настроенной системой сборки и развертывания системы;

## Функциональные требования

В соответствии с требованиями, изложенными выше, и с последующими согласованиями, были предъявлены следующие требования:

* Создание статей;
* Просмотр статей;
* Удаление статей;
* Обновление статей;
* Фильтрация и сортировка статей по признакам, который укажет пользователь;
* Определение отношения статьи к интересующему пользователю классу;

Дополнительные требования:

* Возможность записи статей с локального хранилища в формате .xml в БД веб-сервиса, полученные с помощью приложения «newsGrabb3rForCoursework».

## 2.3 Нефункциональные требования

В соответствии с требованиями, изложенными выше, и с последующими согласованиями, были предъявлены следующие требования:

* Реализуемая система должна быть масштабируемой для обеспечения дальнейшей поддержки и развития проекта
* Реализуемая система должна поддерживать разные режимы работы растеризации страницы браузером (минимальное разрешение – 800х600);
* Реализуемая система должна быть устойчива к некорректным пользовательским запросам;
* Клиентская часть должна быть понятна пользователю.

# 3 АРХИТЕКТУРА ПРИЛОЖЕНИЯ

Приложение будет реализовано по паттерну MVC.

Для разработки непосредственно самого приложения выбрана IDE PyCharm Professional 2021, ввиду большого функционала и удобства работы.

Пример архитектуры проекта представлен на рисунке 0.

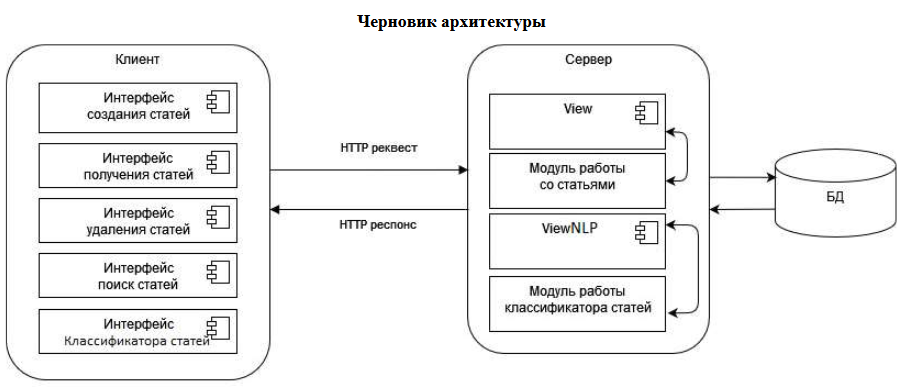


Рисунок 0 – архитектура проекта

Дополнить, много.

# 4 ОПИСАНИЕ КЛАССИФИКАТОРА

## 4.1 Классы

Были выбраны следующие классы:

* Автоновости
* В мире
* В России
* Инопресса
* Культура
* Медицина
* Недвижимость
* Спорт
* Технологии
* Экономика

скорректировать

## 4.2 Вектор признаков

Сумка слов, связанных с темой текстов обучающей выборки

TF-IDF – Надстройка над способом векторизации

дополнить

## 4.3 Обучающие и тестовые коллекции

Составлена из 50 статей для каждой категории, которая так же является и тестовой выборкой. Для выбора статей был написан запрос к БД следующего вида и представлен в листинге 1.

SELECT id FROM XMLCollection\_article where category='category' limit 50;

Листинг 1 – выборка статей из БД

Где параметр «category» соответствует одному из классов, указанных в пункте 1.

Дополнить

# 5 ОПИСАНИЕ ИНФРАСТРУКТУРЫ РАЗРАБОТКИ

Выбранные технологии:

Frontend

1. HTML – язык разметки для веб сайтов.

2. CSS – язык стилей для веб сайтов. Для оформления страницы.

3. Js – популярный скриптовый язык для сайтов. Для сценариев.

Backend:

1. Python 3.6 и Django 3.2 – язык программирования и фреймворк для этого языка.

2. Библиотека scikit-learn для реализации машинного обучения.

3. Встроенная БД SQLite.

Для разработки непосредственно самого приложения выбрана IDE PyCharm Professional 2021, ввиду большого функционала и удобства работы.

Дополнить

# 6 РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ

## 6.1 Сценарии тестирования

### 6.1.2 Автоматическое тестирование клиентской части.

Для автоматического тестирования применим инструмент Postman, с помощью которого будут делаться обращения напрямую к серверу как с корректными данными, так и с некорректными, для получения результатов обращений в рамках протокола HTTP.

Тута мы пишем про POSTman

### 6.1.1 Сценарий ручного тестирования клиентской части

Для ручного тестирования применялись следующие браузеры: Vivaldi, Chrome, Edge, IE11, Firefox, Waterfox, Firefox Nightly. В дальнейшем, для сокращения количество повторяющихся данных, будут писаться результаты вкупе, с ответом “good” – «удовлетворительный результат» и “bad” - «неудовлетворительный результат».

А здесь хрень с баки на 10 страниц

## 6.2 Результаты тестирования

В результате автоматического тестирования не удалось выявить неполадки на этапе загрузки сервера и отправки POST\GET запросов. Случай верного ответа от сервера при верных и неверных запросах получены во всех случаях, тем самым, загружались требуемые страницы\ответы.

В результате ручного тестирования были доработаны недостатки, выявленные при отображении данных на страницах. Таким образом, удалось нивелировать последствия от работы растеризации страниц разными браузерами и в итоге достигнуть качественного результата.

# 7 ИНТЕРФЕЙС И ВОЗМОЖНОСТИ СИСТЕМЫ

Веб-интерфейс приложения условно можно разделить на:

* Страница создания статей (newsru | Создание статьи) – страница, отвечающая за создание статей пользователем и с внесения определённых признаков (таких как заголовок, дата создания, текст статьи, теги, ссылка на источник). Пример интерфейса веб-страницы представлен на рисунке 1.

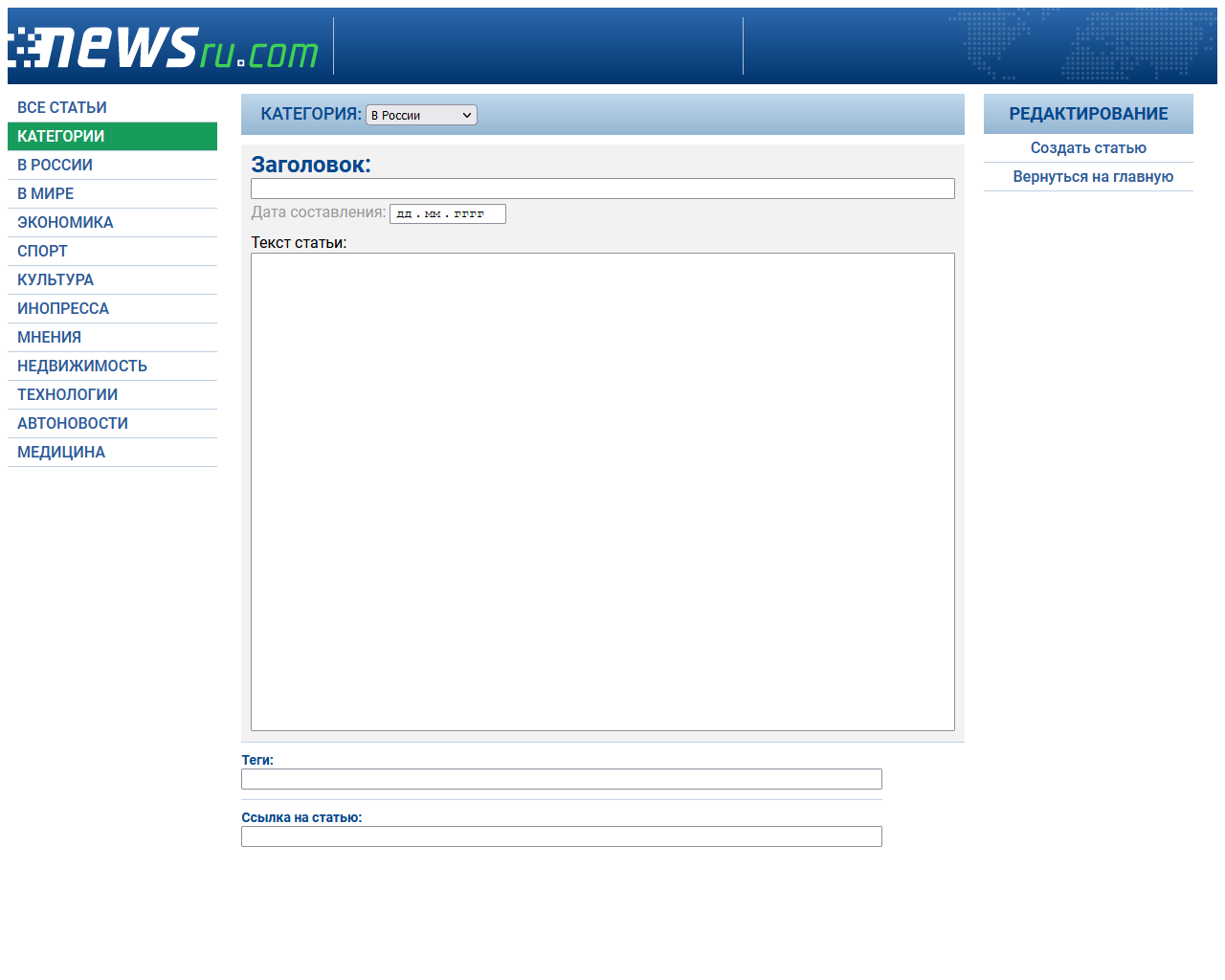


Рисунок 1 – веб-интерфейс страницы «создания статей»

* Главная страница (newsru | Коллекция) – главная страница, на которой можно воспользоваться поиском нужных статей по заданным критериям, узнать результаты классификации текста, перейти на другую страницу со статьями или в другую категорию. Пример интерфейса веб-страницы представлен на рисунке 2.

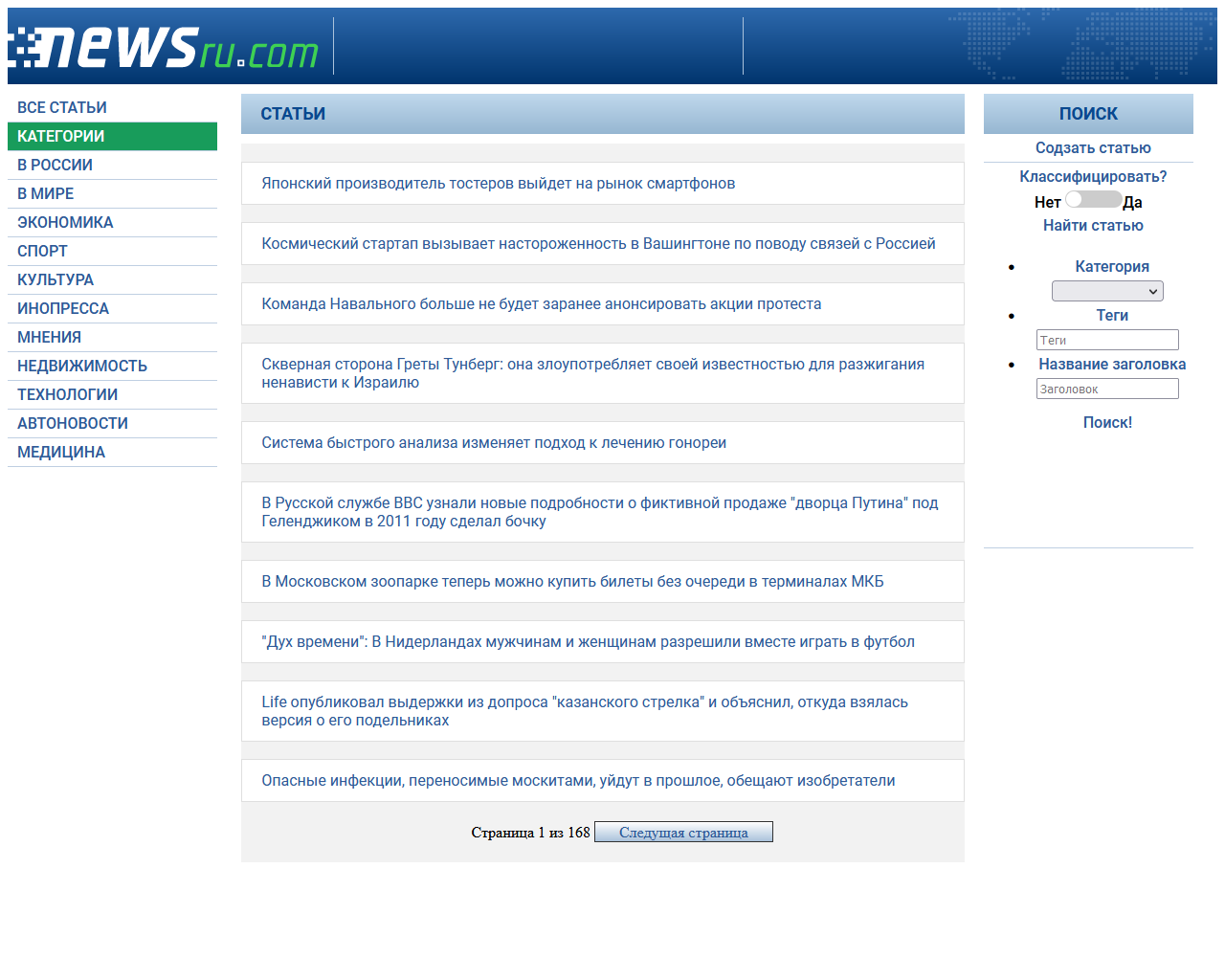


Рисунок 2 – веб-интерфейс страницы «главная страница»

* Страница новости (newsru | 'название новости') – страница, содержащая в себе саму новость. Обладает возможностью редактирования статьи и её удаления. С данной страницы можно так же перейти в другую категорию. Пример интерфейса веб-страницы представлен на рисунке 3.



Рисунок 3 – веб-интерфейс страницы «новость»

* Страницы ошибок (ошибочка 'код ошибки') – служебные страницы. Отображаются в случае ошибок и\или некорректного запроса пользователя. Пример интерфейса веб-страницы представлен на рисунке 4.

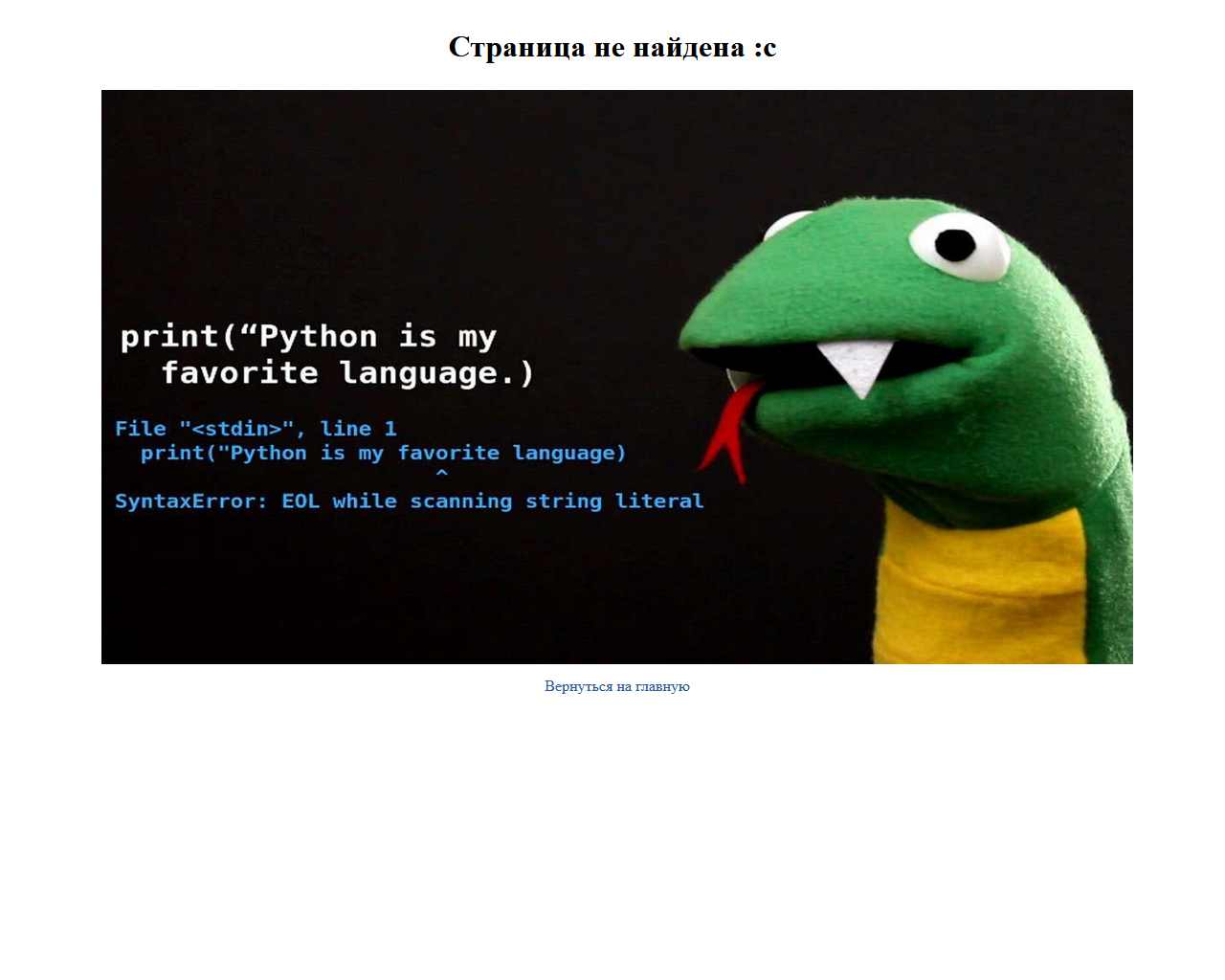


Рисунок 4 – веб-интерфейс страницы «ошибка 404»

Сам веб-интерфейс наследован с сайта [] newsru.com, с которого первоначально забирались статьи. Данный веб-ресурс отличается своей простотой в использовании и интуитивно понятным интерфейсом, а также поддерживает разно размерные устройства – поддержка в форматах txt, palm, pda, classic и без префикса третьего уровня. Интерфейс сайта не менялся с 2009 года. Но главным остаётся тот факт, что интерфейс не претерпевает существенных изменений. За основу был взят интерфейс без префикса третьего уровня.

# 8 АНАЛИЗ РЕЗУЛЬТАТОВ

Я устал, когда уже это всё закончится то?

# СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Проект DjangoCooledByCollection («Django Unchained для коллекций статей с сайта newsru.com. Просмотр, классификация, редактирование».) [Электронный ресурс]. – Режим доступа: github.com/Vadim000009/DjangoCooledByCollection Просмотрено: 20.12.2021

2.

3.