



ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ  
«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ  
(национальный исследовательский университет)» (МАИ)

---

Институт № 3 «Системы управления, информатика и электроэнергетика»  
Кафедра 317 «Управление инновациями»  
27.03.05 «Инноватика»

## **Курсовая работа**

по дисциплине

«Структура программного обеспечения и базы данных»

на тему

«Веб-приложение для формирования отчетов по Лабораторной работе»

Выполнили:

Студенты группы МЗО-234Б-22

Зарубкин Т.П. \_\_\_\_\_

Васильев Я.О. \_\_\_\_\_

Преподаватель:

Александрова С.С. \_\_\_\_\_

Москва

2023

# Оглавление

<b>Введение.....</b>	<b>3</b>
<b>Цель работы .....</b>	<b>3</b>
<b>Описание предметной области .....</b>	<b>3</b>
<b>Обзор инструментов и методов: .....</b>	<b>4</b>
Использование языка программирования Python .....	4
Использование библиотеки Docx .....	5
Использование фреймворка Django .....	6
Использовние SQLite .....	7
Использование Html.....	8
<b>Описание кода .....</b>	<b>10</b>
Настройки и конфигурации.....	10
Алгоритм замены текста в шаблоне .....	15
<b>Внешний вид приложения .....</b>	<b>18</b>
<b>Заключение: .....</b>	<b>23</b>

# **Введение**

В настоящее время множество студентов сталкивается с необходимостью написания отчетов по лабораторным работам, поэтому требуется современное решение проблемы. Это решение должно быть простым и доступным а также помочь создать систему с унифицированными отчетами, чтобы упростить работу как студентов, так и преподавателей.

## **Цель работы**

Цель данной курсовой работы заключается в разработке и реализации веб-приложения, которое позволит студентам создавать отчеты по лабораторным работам в удобном и интуитивно понятном интерфейсе. Приложение должно предоставлять возможность добавления и оформления текстового содержимого, вставки графиков, таблиц, изображений и других элементов, необходимых для полноценного представления результатов исследований.

## **Описание предметной области**

Предметная область данной курсовой работы связана с разработкой веб-приложений с использованием фреймворка Django, языка программирования Python и языка разметки HTML. Django является мощным инструментом для создания веб-приложений на языке Python. Он предоставляет удобные средства для работы с данными, управления URL-адресами и создания визуального представления данных. Python, в свою очередь, является популярным языком программирования, который обладает широкими возможностями и простотой синтаксиса. HTML используется для создания структуры и визуального представления веб-страниц.

В рамках данной работы будет рассмотрено использование Django для создания моделей данных, представлений и управления URL-адресами веб-

приложения. HTML будет применяться для создания пользовательского интерфейса и отображения данных на веб-страницах.

Сочетание Django, Python и HTML позволяет разрабатывать мощные и гибкие веб-приложения с удобным интерфейсом и эффективной обработкой данных.

## **Обзор инструментов и методов:**

### **Использование языка программирования Python**

Python является мощным и популярным языком программирования, который широко используется в различных областях разработки, включая веб-приложения. Он обладает простым и понятным синтаксисом, что делает его доступным для начинающих разработчиков.

Python поддерживает различные парадигмы программирования, включая процедурное, объектно-ориентированное и функциональное программирование. Это позволяет разработчикам выбирать наиболее подходящий подход для своих проектов и создавать гибкий и модульный код.

Одним из главных преимуществ Python является его богатая экосистема библиотек и модулей. Существует множество библиотек, специально разработанных для веб-разработки, таких как Django, Flask и Pyramid. Django, например, предоставляет удобные средства для создания моделей данных, представлений и управления URL-адресами веб-приложений. Flask и Pyramid также предлагают гибкость и расширяемость для разработки веб-приложений на Python.

Кроме того, Python имеет богатую библиотеку для работы с данными, включая NumPy, Pandas и Matplotlib. Они обеспечивают удобные инструменты для анализа данных, визуализации и машинного обучения.

Python также известен своей активной и дружелюбной сообществом разработчиков. Существует множество онлайн-ресурсов, форумов и

документации, которые помогают разработчикам получить поддержку и решить возникающие проблемы.

В целом, использование языка программирования Python в разработке веб-приложений обеспечивает гибкость, простоту и эффективность. Он позволяет разработчикам создавать мощные и функциональные веб-приложения с минимальными усилиями.

## **Использование библиотеки Docx**

Библиотека docx является мощным инструментом для работы с документами в формате Microsoft Word (.docx). Она позволяет разработчикам создавать, редактировать и форматировать документы Word с помощью языка программирования Python.

Использование библиотеки docx дает возможность автоматизировать процесс создания и обработки документов Word. Вы можете создавать новые документы, добавлять текст, таблицы, изображения, графики и другие элементы форматирования.

Библиотека docx также предоставляет функциональность для изменения стилей, шрифтов, цветов и др. Вы можете управлять разметкой документа, разбивать его на разделы, добавлять заголовки, нумерацию страниц, содержание и другие элементы структуры.

Кроме того, библиотека docx поддерживает работу с макросами и формулами, что позволяет вам создавать динамические документы с автоматическими вычислениями и взаимодействием с другими приложениями.

Использование библиотеки docx в разработке может быть полезным во многих областях, таких как создание отчетов, генерация документации, автоматизация процесса создания контента и многое другое.

В целом, библиотека `docx` предоставляет разработчикам удобный и эффективный способ работы с документами Word через Python, что делает ее ценным инструментом для разработки и автоматизации процессов, связанных с документами Word.

## **Использование фреймворка Django**

Django - это высокоуровневый фреймворк для разработки веб-приложений на языке программирования Python. Он предоставляет множество инструментов и методов, которые значительно упрощают процесс создания и развертывания веб-приложений.

Одним из основных преимуществ Django является его модульность. Фреймворк предоставляет набор готовых компонентов, которые можно использовать для различных задач, таких как работа с базами данных, обработка форм, аутентификация пользователей и многое другое. Это позволяет разработчикам значительно ускорить процесс разработки и сосредоточиться на бизнес-логике приложения.

Django также обладает встроенной системой администрирования, которая позволяет разработчикам легко создавать административные интерфейсы для управления данными в приложении. Это упрощает работу с базами данных и позволяет администраторам эффективно управлять содержимым.

Важной особенностью Django является его поддержка шаблонизации. Фреймворк предоставляет удобный язык шаблонов, который позволяет разработчикам разделять логику приложения и представление данных. Это упрощает создание динамических и интерактивных веб-страниц.

Одним из главных преимуществ Django является его мощная система маршрутизации URL. Она позволяет легко настраивать обработчики запросов и определять, какие действия должны быть выполнены для каждого URL-

адреса. Это делает процесс навигации по веб-приложению легким и интуитивно понятным.

Кроме того, Django обладает широкой поддержкой сообщества разработчиков, что позволяет получить помощь и поддержку в решении возникающих проблем. Большое количество документации, учебных материалов и сторонних библиотек делают Django мощным инструментом для разработки веб-приложений.

В целом, Django является мощным и гибким фреймворком для разработки веб-приложений, который предлагает широкий набор инструментов и методов. Он позволяет разработчикам создавать высококачественные веб-приложения с минимальными усилиями и сосредоточиться на реализации бизнес-логики.

## **Использование SQLite**

SQLite - это встроенная реляционная база данных, которая не требует отдельного сервера и управляет данными непосредственно на устройстве, где используется. Плюсы использования SQLite:

1. Легкость использования: SQLite прост в использовании и не требует сложной настройки. Он может быть интегрирован в приложения на разных платформах, включая мобильные устройства и настольные компьютеры.

2. Поддержка SQL: SQLite полностью совместим с языком структурированных запросов SQL, что делает его удобным для работы с данными в реляционной форме. Он предоставляет множество функций и операторов SQL для управления данными.

3. Надежность: SQLite обеспечивает надежность данных благодаря транзакционной обработке и поддержке механизма сохранения данных на диске. Это позволяет избежать потери данных в случае сбоев или сбоев питания.

4. Портативность: Базы данных SQLite представлены в виде одного файла, что делает их легко переносимыми между различными платформами и операционными системами. Это позволяет разработчикам создавать приложения, которые могут без проблем работать на разных устройствах.

5. Эффективность: SQLite предлагает хорошую производительность, особенно для небольших и средних баз данных. Он обеспечивает быстрый доступ к данным и эффективное выполнение запросов.

6. Масштабируемость: SQLite поддерживает работу с множеством одновременных соединений и может обрабатывать базы данных разных размеров. Он также поддерживает индексы, что позволяет ускорить выполнение запросов.

7. Открытый исходный код: SQLite является проектом с открытым исходным кодом, что означает, что его исходный код доступен для изучения и модификации. Это предоставляет возможность разработчикам адаптировать SQLite под свои нужды и вносить улучшения в базу данных.

Стоит отметить, что SQLite имеет свои ограничения, такие как небольшие возможности масштабирования для больших баз данных или отсутствие поддержки некоторых функций, которые могут быть доступны в других реляционных базах данных.

## **Использование Html**

HTML (HyperText Markup Language) - это язык разметки, который используется для создания структуры и визуального представления веб-страниц. Плюсы использования HTML:

1. Простота использования: HTML является простым и понятным языком, который легко изучить и использовать. Он основан на тегах и атрибутах, которые определяют структуру и содержимое веб-страницы.



2. Семантическая разметка: HTML позволяет использовать семантические теги для описания содержимого веб-страницы. Это помогает поисковым системам и другим инструментам понять структуру и смысл информации на странице.

3. Разделение структуры и стиля: HTML отвечает за структуру веб-страницы, в то время как стили и визуальное представление задаются с помощью CSS (Cascading Style Sheets). Это позволяет разделять содержимое и оформление, делая код HTML более чистым и поддерживаемым.

4. Поддержка мультимедиа: HTML предоставляет теги для вставки изображений, видео и аудио на веб-страницу. Это позволяет создавать богатые мультимедийные контенты и улучшать пользовательский опыт.

5. Гиперссылки: HTML позволяет создавать ссылки на другие веб-страницы или ресурсы. Это обеспечивает навигацию между страницами и позволяет пользователям легко перемещаться по сайту.

6. Формы и ввод данных: HTML предоставляет теги для создания форм, которые позволяют пользователям вводить данные и отправлять их на сервер для обработки. Это полезно для создания интерактивных элементов, таких как регистрационные формы, комментарии и другие.

7. Поддержка мобильных устройств: HTML имеет возможность адаптироваться к различным экранам и устройствам, что делает его подходящим для создания адаптивных и мобильных веб-страниц. Это позволяет обеспечить хороший пользовательский опыт на разных устройствах.

В целом, HTML является основным инструментом для создания веб-страниц. Он обладает простым синтаксисом, семантической разметкой и

поддержкой мультимедиа. HTML позволяет создавать структурированный и доступный контент для веб-приложений и сайтов.

## Описание кода

### Настройки и конфигурации

Файл `Settings.py` является ключевым для определения настроек и конфигураций. В этом файле задаются параметры, такие как настройки базы данных, уровень отладки, список установленных приложений, настройки статических файлов, маршрутизация URL, настройки международной локализации и другие параметры, влияющие на поведение веб-приложения.

Этот файл позволяет настраивать поведение и внешний вид своего веб-приложения, а также определять его основные характеристики. Например, здесь можно указать, какая база данных будет использоваться, какие приложения будут включены, как обрабатывать статические файлы, настроить маршрутизацию URL и многое другое.

Используемые

`BASE_DIR = Path(__file__).resolve().parent.parent:` Эта строка устанавливает базовый каталог проекта, определяя родительский каталог текущего файла.

`SECRET_KEY :` Эта строка устанавливает секретный ключ, используемый для криптографической подписи в Django. Важно хранить этот ключ в секрете и не публиковать его публично.

`DEBUG = True:` Эта строка устанавливает режим отладки на `True`, что подходит для разработки. В продакшене он должен быть установлен на `False` в целях безопасности.

`ALLOWED_HOSTS = []:` Эта строка указывает список имен хостов/доменов, которым разрешен доступ к проекту Django. В данном случае список пустой, что означает разрешение доступа любому хосту.

`INSTALLED_APPS`: Это список установленных приложений Django в проекте. Каждое приложение предоставляет определенную функциональность для проекта. В нашем случае включены, например, `'django.contrib.admin'`, `'django.contrib.auth'` и другие приложения.

`MIDDLEWARE`: Это список классов промежуточного программного обеспечения (middleware), которые обрабатывают запросы и ответы. Middleware может выполнять различные функции, такие как аутентификация, безопасность и изменение запроса/ответа.

`ROOT_URLCONF = 'lab_generation.urls'`: Эта строка указывает Python-модуль, содержащий конфигурацию URL-адресов для проекта.

`TEMPLATES`: Это список словарей, в которых определяется конфигурация для шаблонов проекта. Он включает параметры для каталогов шаблонов, контекстных процессоров и других настроек.

`WSGI_APPLICATION = 'lab_generation.wsgi.application'`: Эта строка указывает WSGI-приложение, используемое в проекте.

`DATABASES`: Этот словарь определяет настройки базы данных для проекта. В данном коде используется база данных SQLite с именем `'db.sqlite3'`, расположенной в `BASE_DIR`.

`AUTH_PASSWORD_VALIDATORS`: Это список правил проверки паролей для пользователей. Он включает валидаторы для проверки сходства, длины, общих паролей и числовых паролей.

`LANGUAGE_CODE = 'ru'`: Эта строка устанавливает код языка по умолчанию для проекта на русский.

`TIME_ZONE = 'UTC'`: Эта строка устанавливает часовой пояс по умолчанию для проекта на UTC.

`USE_I18N = True` и `USE_TZ = True`: Эти строки включают поддержку интернационализации и часовых поясов в проекте.

`STATIC_URL`, `STATIC_ROOT` и `STATICFILES_DIRS`: Эти строки определяют настройки для обслуживания статических файлов. `STATIC_URL` определяет префикс URL для статических файлов, `STATIC_ROOT` - каталог, в котором будут собраны статические файлы, а `STATICFILES_DIRS` - список дополнительных каталогов для поиска статических файлов.

`DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'`: Эта строка устанавливает тип поля первичного ключа по умолчанию для моделей в виде большого автоинкрементного поля.

`MEDIA_ROOT` и `MEDIA_URL`: Эти строки определяют настройки для обработки медиа-файлов. `MEDIA_ROOT` указывает каталог, где будут храниться загруженные медиа-файлы, а `MEDIA_URL` - префикс URL для доступа к медиа-файлам.

`AUTHENTICATION_BACKENDS`: Эта строка указывает аутентификационные бэкэнды, используемые для аутентификации пользователей. В данном коде используется стандартный `ModelBackend`.

`LOGIN_REDIRECT_URL`, `LOGIN_URL` и `LOGOUT_URL`: Эти строки указывают URL-адреса для перенаправления после входа, входа и выхода соответственно.

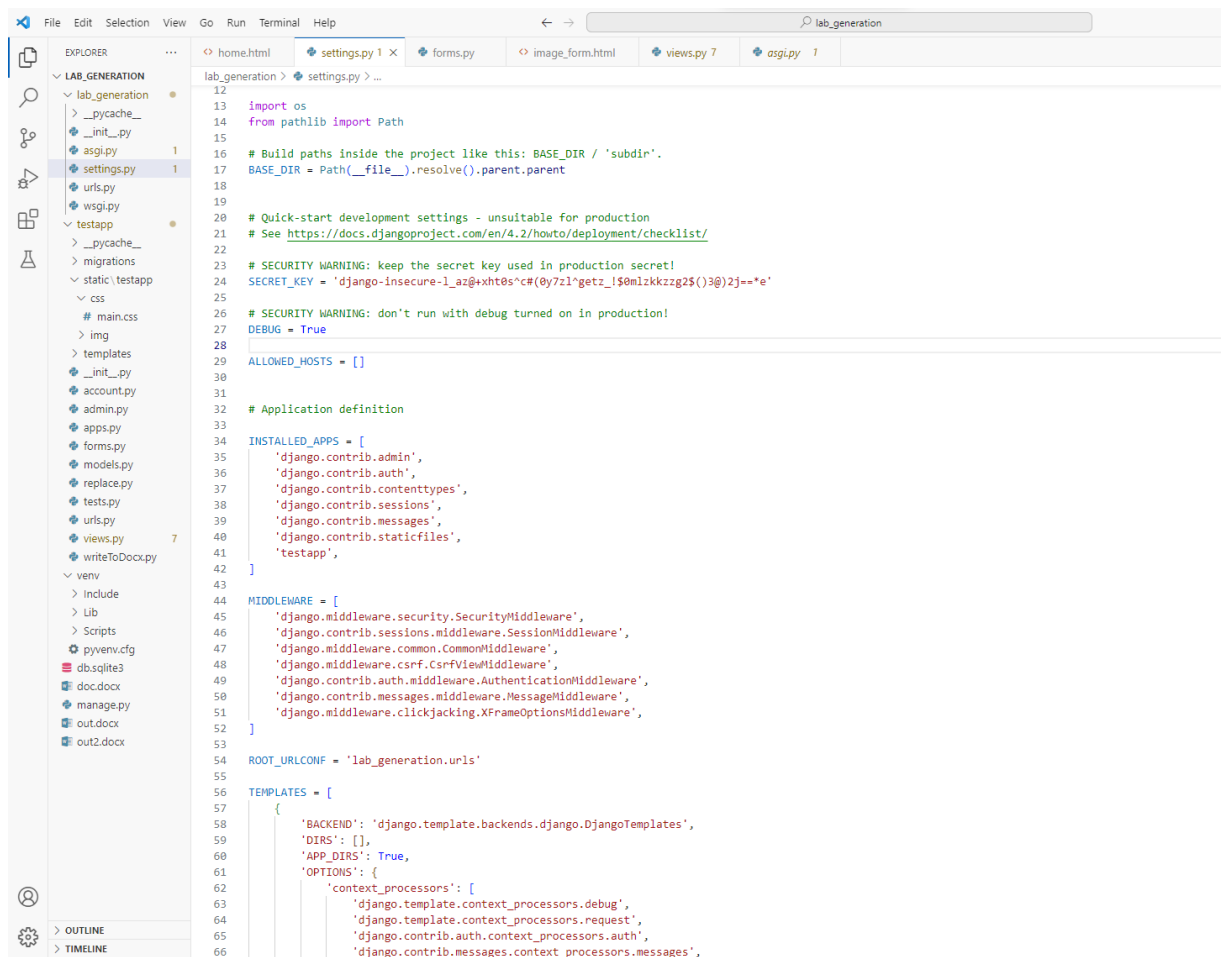


Рисунок 1 код settings.py

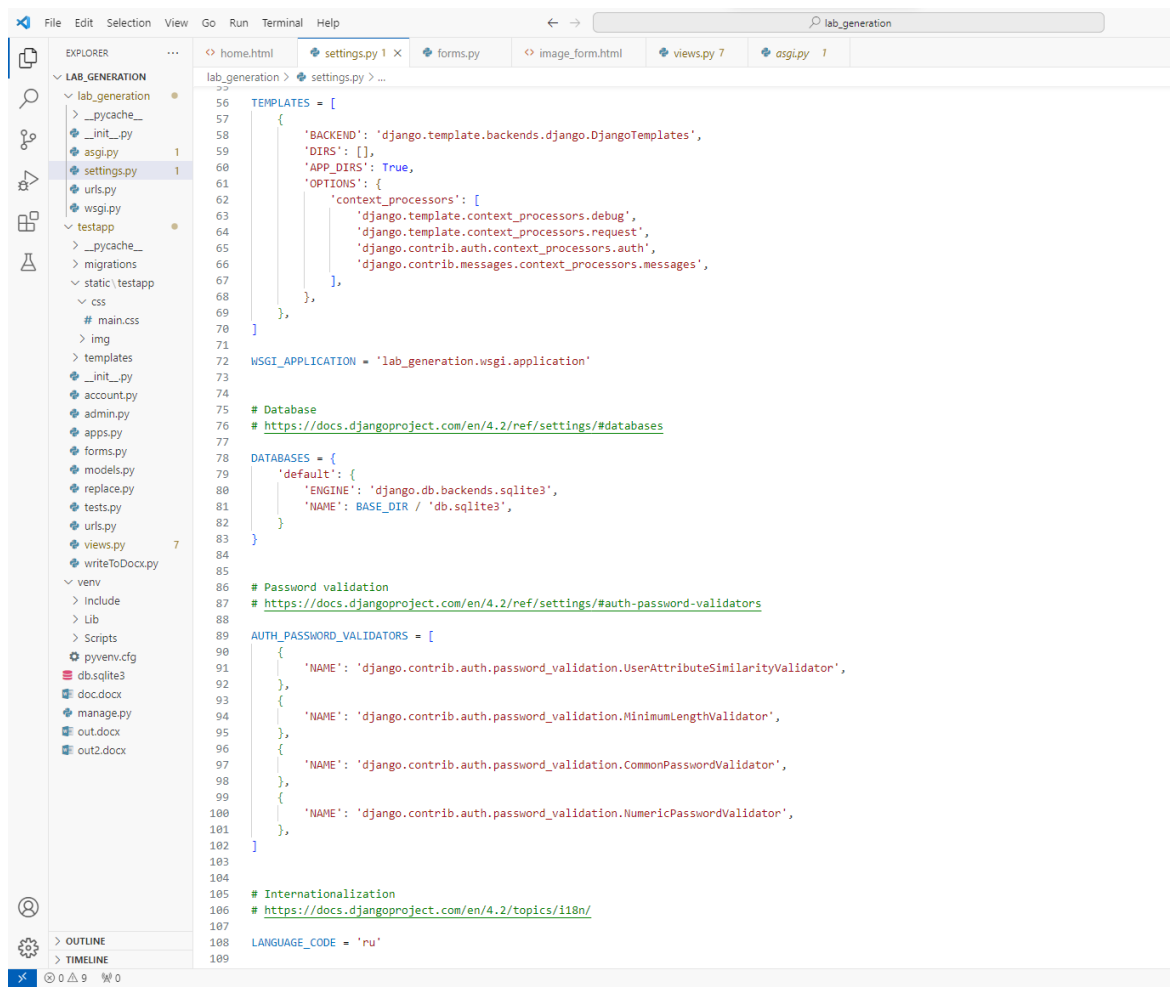


Рисунок 2 код settings.py

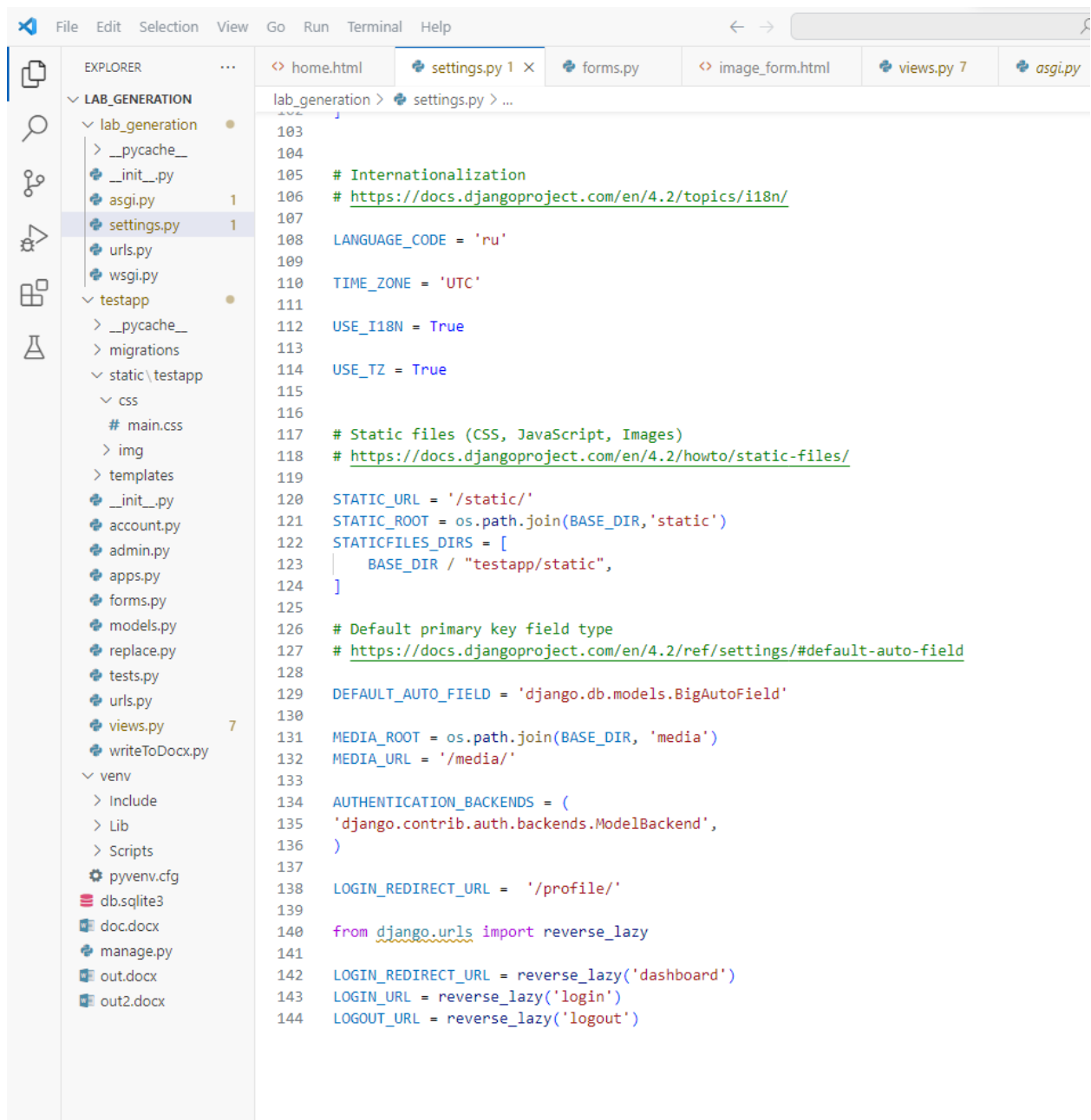


Рисунок 3 код settings.py

## Алгоритм замены текста в шаблоне

Создаем словарь keyword\_dict, который содержит ключевые слова и их значения для замены в документе. Также задаем

переменные `input_file` и `output_file`, которые указывают на входной и выходной файлы соответственно. Функция `checkIfExists()` проверяет, существует ли уже выходной файл. Если выходной файл не существует, то функция открывает входной файл, сохраняет его как выходной файл и закрывает его. Функция `replace()` выполняет замену ключевых слов в документе. Для каждого ключевого слова в `keyword_dict`, функция создает временный документ `tmp.docx`, заменяет текст в выходном файле с использованием `replace_docx_text()`, открывает временный документ, сохраняет его как выходной файл и удаляет временный документ. Функцию `checkIfExists()` проверяет существования выходного файла. Функция `replace()` выполняет замену ключевых слов в документе. Функция `os.remove` удаляет временный файл `tmp.docx`.



```

testapp > replace.py > replace
1  import docx
2  from docx2python.utilities import replace_docx_text
3  import os
4
5  keyword_dict = {
6      "[num]": "",
7      "[group]": "",
8      "[username]": "",
9      "[username1]": "",
10     "[mission]": ""
11 }
12
13 input_file = "doc.docx"
14 output_file = "out.docx"
15
16 def checkIfExists():
17     if not os.path.isfile(output_file):
18         document = docx.Document(input_file)
19         document.save(output_file)
20
21 def replace():
22     for i in keyword_dict:
23         doc = docx.Document()
24         doc.save("tmp.docx")
25         tmp_file = "tmp.docx"
26         replace_docx_text(
27             output_file,
28             tmp_file,
29             (i, keyword_dict[i]),
30             html=True,
31         )
32         doc = docx.Document("tmp.docx")
33         doc.save(output_file)
34     os.remove(tmp_file)
35

```

Рисунок 4 код Алгоритм замены текста в шаблоне

# Внешний вид приложения

Так выглядит сайт, для не вошедших в систему пользователей

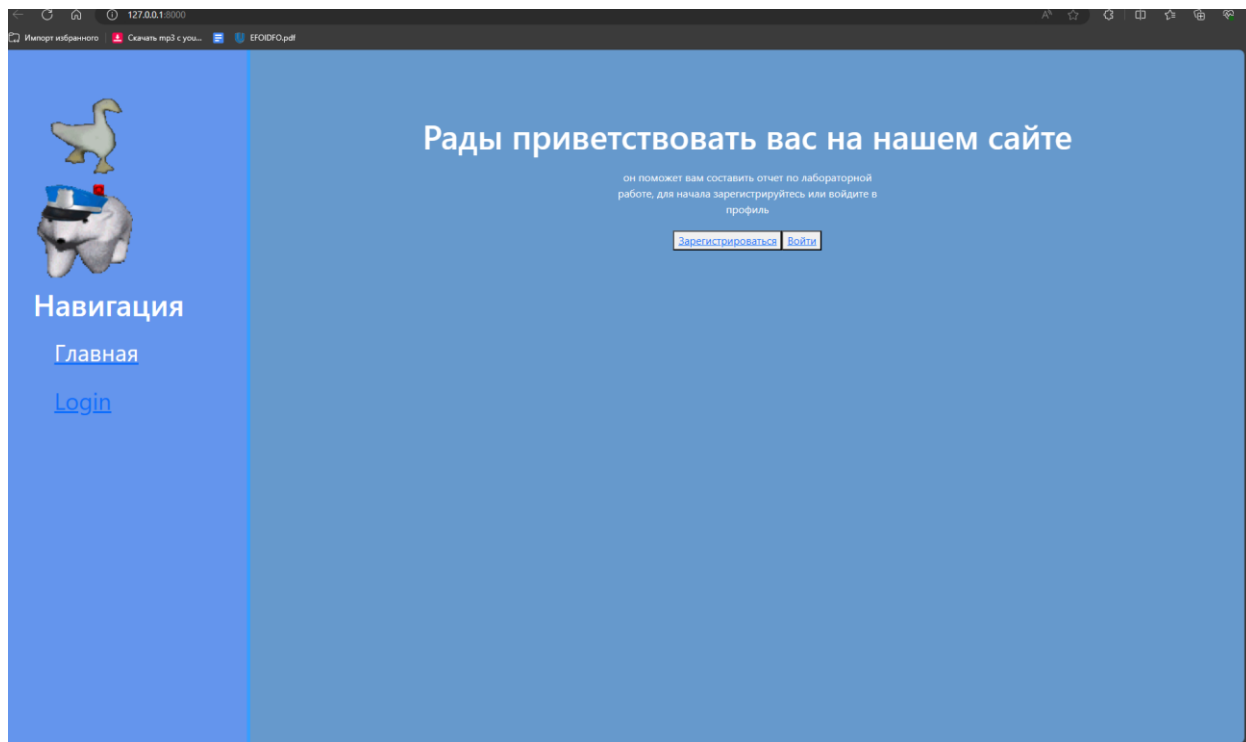


Рисунок 5 главная страница

Внешний вид страницы регистрации. Здесь доступны поля для ввода имени пользователя (login), имени, пароля и подтверждения пароля. Для безопасности введенный пароль будет проверен на надежность, и, если пароль будет ненадежный, пользователь получит сообщение об этом.

Создать аккаунт

Пожалуйста, следуйте следующей форме:

Имя пользователя:  Обязательное поле. Не более 150 символов. Только буквы, цифры и символы @/./+/\_.

Имя:

Пароль:

Повторите пароль:

Навигация

[Главная](#)

[Login](#)

Рисунок 6 страница 'register/'

Так выглядит сайт для людей, вошедших в профиль

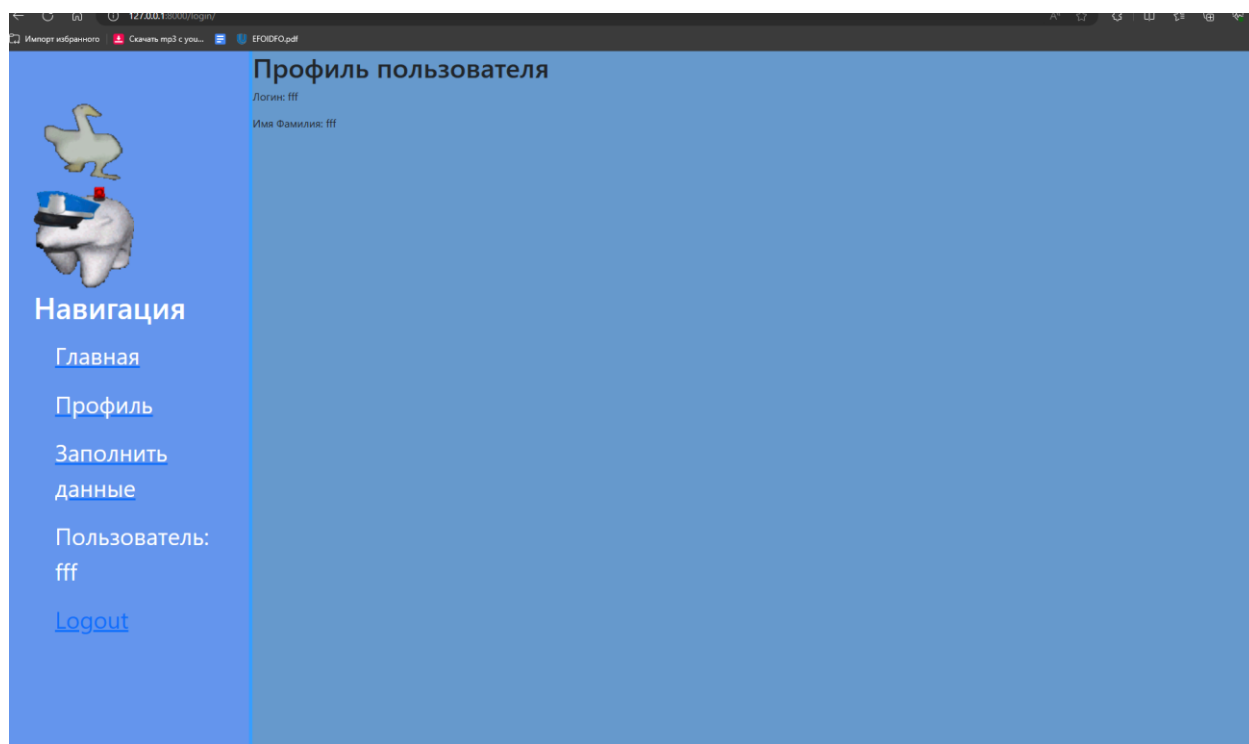



Рисунок 7 страница '/login'

Страница с заполнением личной информацией об учащимся, которая нужна для внесения в отчет по лабораторной работе.

A screenshot of a web browser showing a form titled 'Заполнить данные'. The form contains several input fields: 'Введите номер лабораторной работы:', 'Введите номер группы:', 'Введите ФИО:', and 'Введите ФИО(опционально:'. Below these is a large text area for 'Запишите цель работы:'. At the bottom, there is a field for 'Количество разделов(не более 10):' and a 'Вывод:' section with a 'Далее' button. The left navigation menu is identical to the previous screenshot, with the 'Заполнить данные' link highlighted. The profile picture of the pig and duck remains in the top left corner.

Рисунок 8 страница "generalInfo/"

Страница с заполнением содержания лабораторной работы. В поле chapter мы вводим название раздела, в поле text вводим содержание этого раздела



## Навигация

- [Главная](#)
- [Профиль](#)
- [Заполнить данные](#)
- Пользователь: fff
- [Logout](#)

### Заполнить главы

Глава 1Глава 2

**Chapter:** Работа с джанго

**Text:** Здесь какой-то очень важный текст, который идеально подходит в эту лабораторная работа. В нем раскрываются важные аспекты затронутые темой лабораторной работы и написан код с комментариями

**Chapter:** работа с ски

**Text:** Здесь несомненно тоже какой-то важный текст, который будет в тему лабораторной работы и ответит на все вопросы, которые предполагались лабораторной работой

Готово

## Результат работы программы

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ  
«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ  
(национальный исследовательский университет)» (МАИ)

Направление подготовки: 27.03.05 «Инноватика»

### Лабораторная работа №1

по дисциплине «Структура программного обеспечения и базы данных»

Выполнил:

Студенты гр. 12

Васильев Ярослав Олегович

Зарубкин Тимофей Петрович

Преподаватель:

Александрова С.С.

Москва 2022

Рисунок 9 страница 1-ая в готовом отчете

## Цель работы: ффф.

### Работа с джанго

Здесь какой-то очень важный текст, который идеально подходит в эту лабораторная работа. В нем раскрываются важные аспекты затронутые темой лабораторной работы и написан код с комментариями

### работа с скл

Здесь несомненно тоже какой-то важный текст, который будет в тему лабораторной работы и ответит на все вопросы, которые предполагались лабораторной работой

### Вывод

В ходе лабораторной работы мы сделали все что нам нужно и мы очень довольны этим ^\_^

Рисунок 10 страница 2-ая в готовом отчете

## Заключение:

В ходе выполнения курсовой работы мы разработали и реализовали веб-приложение, которое помогает студентам создавать отчеты по лабораторным работам. Также мы закрепили навыки, полученные за время обучения, применив их для создания данного приложения.

Возможные улучшения: в дальнейшем планируется добавить в приложение возможность добавления и оформления текстового содержимого, вставки графиков, таблиц, изображений и других элементов, необходимых для полноценного представления результатов исследований.