МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет компьютерных технологий и прикладной математики

Кафедра вычислительных технологий

**ОТЧЕТ О ПРОХОЖДЕНИИ ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ (ПРЕДДИПЛОМНАЯ ПРАКТИКА)**

по направлению подготовки

02.03.02 Фундаментальная информатика и информационные технологии

период с 29.04.2024 г. по 15.05.2024 г

Выполнил студент \_\_\_46\_ гр.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_Гончаренко В. В. \_\_\_\_\_\_\_\_\_\_

*(подпись) (Ф.И.О. студента)*

Руководитель практики (**производственная практика (преддипломная практика)**)

к.т.н., доцент Приходько Т.А.

*(ученое звание, должность) (подпись) (Ф.И.О)*

Оценка по итогам защиты практики: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_15\_\_» \_\_\_\_мая\_\_\_\_\_\_ 2024 г.

Краснодар

2024 г.

ФГБОУ ВО «КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет компьютерных технологий и прикладной математики

Кафедра вычислительных технологий

**ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ, ВЫПОЛНЯЕМОЕ В ПЕРИОД**

**ПРОВЕДЕНИЯ   
ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ**

**(ПРЕДДИПЛОМНАЯ ПРАКТИКА)**

Студент Гончаренко Валентина Викторовна\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(фамилия, имя, отчество полностью)*

Направление подготовки 02.03.02 Фундаментальная информатика и информационные технологии

Место прохождения практики \_\_\_ ФГБОУ ВО «Кубанский государственный университет»\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Срок прохождения практики с 29.04.2024 г. по 15.05.2024 г

Целью преддипломной практики является формирование и развитие профессиональных знаний в сфере избранной специальности, овладение необходимыми компетенциями по избранному направлению специализированной подготовки, развитие навыков самостоятельной работы, разработка и апробация оригинальных научных предложений и идей, используемых при подготовке магистерской диссертации, овладение современным инструментарием науки для поиска и интерпретации информации с целью ее использования в процессе разработки, реализации и исследования математических и информационных моделей.

Формирование компетенций, регламентируемых ФГОС ВО:

| **Код компетенции** | **Содержание компетенции (или её части)** | **Планируемые результаты при прохождении практики** |
| --- | --- | --- |
| **ОПК-1** | Способен применять фундаментальные знания, полученные в области математических и (или) естественных наук, и использовать их в профессиональной деятельности | Научился применять фундаментальные знания, полученные в области математических и (или) естественных наук |
| **ОПК-2** | Способен применять компьютерные/суперкомпьютерные методы, современное программное обеспечение, в том числе отечественного происхождения, для решения задач профессиональной деятельности | Может применять компьютерные/суперкомпьютерные методы, современное программное обеспечение для решения задач профессиональной деятельности |
| **ОПК-3** | Способен к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям | Получил навык разработки алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей |
| **ПК-1** | Способен понимать и применять в научно-исследовательской и прикладной деятельности современный математический аппарат, основные законы естествознания, современные языки программирования и программное обеспечение; операционные системы и сетевые технологии | Умеет понимать и применять в научно-исследовательской и прикладной деятельности современный математический аппарат. |
| **ПК-2** | Способен проводить под научным руководством локальные исследования на основе существующих методов в конкретной области профессиональной деятельности | Может проводить под научным руководством локальные исследования на основе существующих методов в конкретной области профессиональной деятельности |
| **ПК-5** | Способен применять в профессиональной деятельности современные языки программирования и методы параллельной обработки данных, операционные системы, электронные библиотеки и пакеты программ, сетевые технологии | Умеет применять в профессиональной деятельности современные языки программирования и методы параллельной обработки данных, операционные системы, электронные библиотеки и пакеты программ, сетевые технологии |

Перечень вопросов (заданий, поручений) для прохождения практики:

1.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ознакомлен \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*подпись студента расшифровка подписи (ФИО)*

Руководитель производственной практики

(научно-исследовательская работа)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Приходько Т. А.

**План-график выполнения работ**:

|  |  |  |  |
| --- | --- | --- | --- |
| № | Этапы работы (виды деятельности) при прохождении практики | Сроки | Отметка руководителя практики о выполнении (подпись) |
| 1 | Инструктаж по технике безопасности, охраны труда, пожарной безопасности, а также правилами внутреннего распорядка обучающихся.  Выбор и обоснование темы исследования. | 29.04.2024 |  |
| 2 | Обзор литературы и других источников. | 30.04.2024-02.05.2024 |  |
| 3 | Разработка и тестирование … | 03.05.2024-08.05.2024 |  |
| 4 |  | 09.05.2024-11.05.2024 |  |
| 5 | Оформление результатов проведенного исследования и их согласование с руководителем (составление отчета о прохождении учебной практики). | 12.05.2024-14.05.2024 |  |
| 6 | Защита отчета. | 15.05.2024 |  |

Ознакомлен \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Гончаренко В. В.\_\_\_\_\_\_\_\_\_\_\_

*(подпись студента) ( расшифровка подписи)*

«29» апреля 2024 г.

Руководитель производственной практики (преддипломная практика)

доцент кафедры вычислительных технологий

факультета компьютерных технологий

и прикладной математики, к.т.н. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Приходько Т.А.

*(подпись) (Ф.И.О. руководителя)*

\*Инструктаж по ознакомлению с требованиями охраны труда, техники безопасности, а также правилами внутреннего трудового распорядка.

**ОЦЕНОЧНЫЙ ЛИСТ**

**результатов прохождения   
ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ**

**(ПРЕДДИПЛОМНАЯ ПРАКТИКА)**

**по направлению подготовки**

**02.03.02 Фундаментальная информатика и информационные технологии**

Фамилия И.О студента Соколов Д.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Курс \_\_\_\_4\_\_\_\_

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | ОБЩАЯ ОЦЕНКА  (отмечается руководителем практики) | Оценка | | | |
| 5 | 4 | 3 | 2 |
|  | Уровень подготовленности студента к прохождению практики |  |  |  |  |
|  | Умение правильно определять и эффективно решать основные задачи |  |  |  |  |
|  | Степень самостоятельности при выполнении задания по практике |  |  |  |  |
|  | Оценка трудовой дисциплины |  |  |  |  |
|  | Соответствие программе практики работ, выполняемых студентом в ходе прохождении практики |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | СФОРМИРОВАННЫЕ В РЕЗУЛЬТАТЕ УЧЕБНОЙ ПРАКТИКИ КОМПЕТЕНЦИИ  (отмечается руководителем практики от университета) | Оценка | | | |
| 5 | 4 | 3 | 2 |
|  | ОПК-1 Способен применять фундаментальные знания, полученные в области математических и (или) естественных наук, и использовать их в профессиональной деятельности |  |  |  |  |
|  | ОПК-2 Способен применять компьютерные / суперкомпьютерные методы, современное программное обеспечение, в том числе отечественного происхождения, для решения задач профессиональной деятельности |  |  |  |  |
|  | ОПК-3 Способен к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям |  |  |  |  |
|  | ПК-1 Способен понимать и применять в научно-исследовательской и прикладной деятельности современный математический аппарат, основные законы естествознания, современные языки программирования и программное обеспечение; операционные системы и сетевые технологии |  |  |  |  |
|  | ПК-2 Способен проводить под научным руководством локальные исследования на основе существующих методов в конкретной области профессиональной деятельности |  |  |  |  |
|  | ПК-5 Способен применять в профессиональной деятельности современные языки программирования и методы параллельной обработки данных, операционные системы, электронные библиотеки и пакеты программ, сетевые технологии |  |  |  |  |

Руководитель практики \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_ Приходько Т.А.\_\_\_\_\_\_\_\_\_\_\_\_

*(подпись) (расшифровка подписи)*

Сведения о прохождении инструктажа по ознакомлению с требованиями охраны труда, технике безопасности, пожарной безопасности, а также правилами внутреннего трудового распорядка

Предприятие Федеральное государственное бюджетное образовательное учреждение высшего образования «Кубанский государственный университет»

Факультет компьютерных технологий и прикладной математики

Кафедра вычислительных технологий

|  |  |
| --- | --- |
| Студент |  |
|  | (ФИО, возраст) |

Дата 29 апреля 2024 г.

**1.     Инструктаж по требованиям охраны труда**

|  |  |
| --- | --- |
| Провел | доцент Приходько Т. А. |
|  | (должность, ФИО сотрудника, проводившего инструктаж, подпись) |

|  |  |
| --- | --- |
| Прослушал | Гончаренко В. В. |
|  | (ФИО, подпись студента) |

**2.     Инструктаж по технике безопасности**

|  |  |
| --- | --- |
| Провел | доцент Приходько Т. А. |
|  | (должность, ФИО сотрудника, проводившего инструктаж, подпись) |

|  |  |
| --- | --- |
| Прослушал | Гончаренко В. В. |
|  | (ФИО, подпись студента) |

**3.     Инструктаж по пожарной безопасности**

|  |  |
| --- | --- |
| Провел | доцент Приходько Т. А. |
|  | (должность, ФИО сотрудника, проводившего инструктаж, подпись) |

|  |  |
| --- | --- |
| Прослушал | Гончаренко В. В. |
|  | (ФИО, подпись студента) |

**4. Инструктаж по правилам внутреннего трудового распорядка**

|  |  |
| --- | --- |
| Провел | доцент Приходько Т. А. |
|  | (должность, ФИО сотрудника, проводившего инструктаж, подпись) |

|  |  |
| --- | --- |
| Прослушал | Гончаренко В. В. |
|  | (ФИО, подпись студента) |

**Отзыв руководителя**

**О ПРОХОЖДЕНИИ ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ**

**(ПРЕДДИПЛОМНАЯ ПРАКТИКА)**

|  |  |
| --- | --- |
| Студент |  |
|  | *(фамилия, имя, отчество полностью)* |

Направление подготовки 02.03.02. Фундаментальная информатика и информационные технологии

За время прохождения производственной практики (Преддипломная практика) студент собрал необходимые для исследования данные, разработал сервер, и на их основе написал заключительные главы дипломной работы. Так же было разработано и протестировано \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

В процессе выполнения практики были поставлены и выполнены следующие задачи:

1.

2.

3.

Были проведены все необходимые мероприятия, связанные с выделением целей и задач ПП, а также с подготовкой её аналитической части. В начале исследования была обоснована актуальность выбранной темы. Все пункты индивидуального плана-графика выполнены в полной мере.

В ходе работы индивидуальное задание выполнено в полном объеме, студент проявил высокий уровень самостоятельности, и творческий подход к его выполнению. Представлен оформленный текст собранного материала. Программа ПП полностью соответствует отчёту. Результаты практики оцениваю на оценку «\_\_\_\_\_\_\_\_\_\_\_».

|  |  |  |
| --- | --- | --- |
| Руководитель студента |  |  |
|  | *(подпись)* | *(расшифровка подписи)* |

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 9](#_Toc164106551)

[Техническое задание 11](#_Toc164106552)

[1 Паттерн MTV в Django 11](#_Toc164106553)

[2 Расположение статических файлов в проекте Django 14](#_Toc164106554)

[3 Сравнение методов представлений с классами представлений. CBV. 17](#_Toc164106555)

[ЗАКЛЮЧЕНИЕ 23](#_Toc164106556)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 24](#_Toc164106557)

# ВВЕДЕНИЕ

С развитием интернет-технологий и ростом спроса на веб-приложения возникает необходимость в использовании эффективных инструментов для их создания. Фреймворк Django, разработанный на языке программирования Python, стал одним из ведущих решений для создания мощных и масштабируемых веб-приложений. Его гибкая архитектура и обширный функционал делают Django предпочтительным выбором для разработчиков по всему миру.

В рамках научно-исследовательской работы была поставлена задача разработать комплексное веб-приложение, предназначенное для оптимизации процесса обмена файлами между преподавателями и студентами в учебных целях.

Актуальность данной работы обуславливается тем, что исследование основных аспектов архитектуры и разработки веб-приложений на Django позволит глубже понять основные принципы, лежащие в основе этого фреймворка, и применять их на практике для создания качественных и масштабируемых веб-приложений.

Основная цель данной научно-исследовательской работы заключается в исследовании основных аспектов архитектуры и разработки веб-приложений с использованием фреймворка Django. Мы рассмотрим ключевые концепции паттерн MTV (Model-Template-View), организацию статических файлов в проекте, а также сравнение методов представлений с использованием функциональных и классовых подходов.

Для реализации поставленной цели предполагается решить следующие задачи:

* изучить принципы работы и основные аспекты архитектуры паттерна MTV (Model-Template-View);
* разобрать организацию хранения статических файлов в проекте;
* сравнить реализацию методов и классов представлений;
* рассмотреть CBV (class-based-views)

Объектом исследования в работе являются архитектура и особенности разработки веб-приложений на фреймворке Django.

Предметом исследования являются паттерн MTV, расположение статических файлов в проекте Django, сравнение методов представлений с классами представлений (CBV).

Информационная база исследования включает в себя официальную документацию Django, академические статьи и публикации, руководства по разработке на Django, опыт работы и анализ проектов, созданных на Django.

В качестве методов исследования использовались анализ литературы и документации, сравнительный анализ различных подходов и практик, экспериментальное тестирование различных методов и практик в разработке веб-приложений на Django.

Научная новизна работы заключается в основанном на официальной документации глубоком анализе архитектурных особенностей фреймворка Django.

Теоретическая и практическая значимость работы состоит в обогащении теоретических знаний о разработке и предоставлении практических рекомендации по выбору оптимальных методов и практик.

Понимание основных принципов и практик разработки на Django поможет разработчикам создавать эффективные, надежные и безопасные веб-приложения, отвечающие современным требованиям веб-разработки.

# Техническое задание

Создать веб-приложение, позволяющее преподавателям и студентам эффективно обмениваться учебными материалами.

К основным функциональным требованиям относится:

* Защита доступа к функционалу приложения с помощью системы аутентификации и авторизации.
* Возможность для преподавателей загружать на сервер файлы, предназначенные для обмена с определенными группами студентов.
* Студенты должны иметь возможность просматривать/скачивать предназначенные им файлы.

Нефункциональные требования:

* Приложение должно обеспечивать быструю загрузку и отображение при любом количестве пользователей.
* Необходимо обеспечить защиту конфиденциальности данных пользователей.
* Приложение должно поддерживать загрузку и отображение различных типов файлов, таких как документы, изображения, аудио и видео.

Для реализации данного приложения был использован следующий технологический стек: язык программирования Python, обеспечивающий высокий функционал фреймворк Django, реляционная и масштабируемая база данных PostgreSQL, HTML и CSS для разработки пользовательского интерфейса.

# Паттерн MTV в Django

Паттерн MTV (Model-Template-View) – ключевой архитектурный принцип, лежащий в основе фреймворка Django. Он помогает разделить логику приложения на три основных компонента: модель (Model), шаблон (Template) и представление (View).

Модель представляет собой абстракцию данных приложения и отвечает за взаимодействие с базой данных. Она определяет поля и их типы, описывающие различные атрибуты объектов, а также взаимосвязи между различными объектами. Модели обеспечивают возможность выполнения валидации данных перед сохранением их в базе данных, что позволяет контролировать целостность данных и обеспечивать их корректность, повышая надежность приложения.

В Django модели представляют собой классы Python, которые наследуются от базового класса django.db.models.Model, они определяются в файле models.py внутри приложения.

Шаблон (Template) в фреймворке определяет внешний вид и форматирование данных для отображения пользователю. Он использует специальный язык шаблонов Django (DTL), который позволяет вставлять динамические данные и логику отображения. Шаблоны обеспечивают разделение логики приложения от представления данных пользователю, что делает код более чистым и поддерживаемым.

Django позволяет использовать наследование шаблонов для создания их иерархии, что позволяет избежать дублирования кода и упрощает поддержку приложения. Родительский шаблон может содержать общую структуру страницы, а дочерние могут расширять его и добавлять специфический контент.

Шаблоны в Django обычно представляют собой файлы с расширением .html, которые хранятся в специальной директории внутри приложения, они могут включать в себя ссылки на статические файлы, такие как CSS-стили, JavaScript-скрипты и изображения.

Представление (View) в Django отвечает за обработку запросов от пользователей, взаимодействие с моделями и определяет, какие данные будут переданы в шаблон для отображения.

Представления принимают HTTP-запросы от клиентов и определяют, какие действия должны быть выполнены в ответ на эти запросы. Это может включать получение данных из базы данных, обработку форм, выполнение вычислений и другие операции, они формируют HTTP-ответы, которые отправляются обратно клиентам. Это может быть HTML-страница, JSON-ответ, файл или другой тип данных, в зависимости от типа запроса и требований приложения.

Представления в Django обычно определяются в файле views.py внутри приложения, они могут быть реализованы как функции или классы. Функциональные представления представляют собой обычные функции Python, которые принимают запрос в качестве аргумента и возвращают ответ. Классовые представления (CBV) представляют собой классы Python, которые содержат методы для обработки различных типов запросов.

В целом, представления в Django являются центральным элементом приложения, определяя его поведение и взаимодействие с клиентами.

Взаимодействие между ключевыми компонентами паттерна MTV определяется следующим образом:

1. Пользователь отправляет HTTP-запрос к веб-приложению, например, запрашивая страницу в браузере.
2. Представление, соответствующее URL-адресу, который был запрошен, получает этот запрос. Представление содержит логику, определяющую, как должен быть обработан этот запрос.
3. В процессе обработки запроса представление может взаимодействовать с моделью, чтобы получить необходимые данные.
4. После получения данных из базы данных модель возвращает их представлению.
5. Представление подготавливает их для отображения на веб-странице. Оно может добавлять дополнительную логику обработки данных или фильтрации перед передачей их в шаблон.
6. Шаблон получает данные от представления и использует их для генерации HTML-кода, который будет отображаться на веб-странице.
7. Сформированный HTML-код шаблона отправляется в качестве HTTP-ответа обратно пользователю, который может видеть отображенные данные на веб-странице в своем браузере.

Изображение выглядит как текст, диаграмма, Шрифт, снимок экрана

Автоматически созданное описание

Рисунок 1 – иллюстрация работы паттерна MTV

# Расположение статических файлов в проекте Django

Приложение, написанное с помощью фреймворка Django, может работать в двух режимах: в режиме отладки (на тестовом веб-сервере) и в режиме эксплуатации (на реальном веб-сервере).

В режиме отладки статические файлы Django ищет во всех подкаталогах приложений. Если приложений несколько – во всех подкаталогах будут искаться статические файлы. К тому же для хранения статических файлов могут быть прописаны нестандартные пути static, где тоже могут храниться статические файлы.

Но в режиме эксплуатации реальные веб серверы будут брать все статические фалы из папки «static», расположенной в каталоге всего проекта.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 2 – возможные расположения папки static

Для подготовки проекта к эксплуатации в терминале проекта необходимо воспользоваться командой «python manage.py collectstatic», после которой из всех приложений содержимое папок static переместится в общую папку.

Чтобы данный функционал корректно работал в пакете конфигурации необходимо определить три следующие константы:

1. STATIC\_URL – префикс URL-адреса для статических файлов, которые подключают в шаблонах.
2. STATIC\_ROOT – путь к общей статической папке, используемой реальным веб-сервером.
3. STATICFILES\_DIRS – список дополнительных (нестандартных) путей к статическим файлам, используемых для сбора и для режима отладки.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 3 – определение констант статических файлов в settings.py

Так как в Django каждая самостоятельная часть сайта представляется в виде отдельного приложения, организовывая в каждом папку static, может случиться конфликт имен. Чтобы избежать данной ситуации внутри папки static необходимо расположить подкаталог с именем разрабатываемого приложения, исполняющий роль пространства имен. И уже в данном каталоге можно создавать подкаталоги для хранения css, js и images файлов.

Иллюстрация расположения статических файлов:

├───coolsite

│ ├───coolsite

│ │ └───\_\_pycache\_\_

│ │ ├───asgi.py

│ │ ├───settings.py

│ │ ├───urls.py

│ │ └───wsgi.py

│ ├───media

│ └───women

│ ├───migrations

│ ├───static

│ │ └───women

│ │ ├───css

│ │ ├───images

│ │ └───js

│ ├───templates

│ ├───templatetags

│ ├───admin.py

│ ├───apps.py

│ ├───forms.py

│ ├───models.py

│ └───views.py

└───venv

Для использования статических файлов в шаблонах предложения в самом начале файла .html необходимо указать тег {% load static %}.

Чтобы подключить конкретный элемент в разметку, сначала нужно прописать вышесозданный тег.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 4 – пример использования статических файлов

Чтобы подключить конкретный элемент в разметку, сначала нужно прописать выше созданный тег.

# Сравнение методов представлений с классами представлений. CBV.

Методы представлений веб-приложений на основе фреймворка Django могут быть реализованы двумя основными способами: функциональные представления и классы представлений.

Функциональные представления – функции языка Python, принимающие в качестве аргумента HTTP-запрос, и возвращающие HTTP-ответ. Такой вид представлений является наиболее простым определением логики обработки запроса в Django.

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

Рисунок 5 – пример простого функционального представления

К плюсам определения представлений через функции можно отнести следующие пункты:

1. Возможность напрямую управлять процессом обработки запросов без необходимости использования объектно-ориентированных конструкций.
2. Данные представления наиболее понятны для начинающих разработчиков.
3. В некоторых ситуациях представления требуют меньше кода для реализации.

Однако есть и минусы:

1. При добавлении новой функциональности в приложение функциональные представления могут стать более сложными в понимании.
2. При работе с повторяющейся функциональностью оказываются менее расширяемыми в сравнении с классами представлений.

Классы представлений – классы, реализующие специальные методы для обработки различных типов HTTP-запросов. Классы – более гибкий и расширяемый подход к определению логики представлений.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 6 – пример реализации представлений через классы

Плюсы организации представлений с помощью классов:

1. Обеспечение более строгого контроля над различными типами HTTP-запросов (GET, POST, PUT и DELETE).
2. Использование классов позволяет лучше организовывать код, разделяя его на модули.
3. Возможность повторного использования кода для легкого управления повторяющейся функциональностью.
4. Наличие базовых классов представлений.

Минусы:

1. В простых случаях представлений классы требуют больше кода для реализации.
2. Возможность более сложного понимания из-за объектно-ориентированной природы.

Одним из преимуществ представлений через классы являются базовые классы, представляющие собой заранее определенные классы, которые предоставляют основную функциональность для работы с представлениями веб-приложений. Эти классы являются основой для создания пользовательских представлений и предоставляют широкий набор методов и атрибутов для управления процессом обработки запросов.

Одним из основных базовых классов представлений является View. Этот класс обеспечивает базовый функционал для обработки HTTP-запросов и является основой для создания пользовательских классов представлений. Он содержит методы, такие как get(), post(), put(), delete() и другие, которые используются для обработки соответствующих типов запросов.

Кроме того, фреймворк Django предоставляет различные базовые классы представлений для выполнения специфических задач, таких как отображение списка объектов (ListView), отображение детальной информации об объекте (DetailView), создание нового объекта (CreateView), обновление объекта (UpdateView) и удаление объекта (DeleteView). Каждый из этих классов предоставляет определенную функциональность и методы для работы с соответствующими операциями CRUD (Create, Read, Update, Delete).

Рассмотрим реализацию представления для отображения списка через функцию и класс, который наследуется от базового класса ListView:

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 7 – реализация представления для главной страницы через функцию

Внутри функции определяется список статей (posts), формируется контекст, который включает этот список, а также другие необходимые данные, такие как меню и заголовок страницы.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 8 – реализация представления для главной страницы через класс

В классе определены атрибуты модели Record, используемой для получения списка статей, шаблон для отображения страницы (template\_name), а также контекстный объект posts, который будет доступен в шаблоне. Метод get\_context\_data используется для формирования контекста.

Класс RecordHome и функция index выполняют одну и ту же функциональность - отображение списка опубликованных статей на главной странице веб-приложения.

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Рисунок 9 – страница, использующая представления index/RecordHome

Рассмотрим реализацию представления для отображения детальной страницы объекта через функцию и класс, наследуемый от DetailView:

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Рисунок 10 – реализация представления через класс

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Рисунок 11 – реализация представления через функцию

Класс ShowPost и функция show\_post выполняют одну и ту же функциональность – отображение конкретной страницы статьи веб-приложения.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок 12 – страница, полученная через ShowPost/show\_post

Можно заметить, что в обоих примерах присутствует повторение кода. Однако, при использовании представлений, определенных с использованием классов, такое повторение может быть организовано с помощью миксинов, что позволяет сократить объем кода. В случае использования функций формирование контекста может стать более сложным и менее явным.

# ЗАКЛЮЧЕНИЕ

В данной работе были рассмотрены некоторые из ключевых аспектов разработки веб-приложений на основе фреймворка Django. Был изучен паттерн MTV – основной архитектурный принцип фреймворка, рассмотрена каждая его компонента: модель, шаблон и представление. Изучена правильная организация хранения статических ресурсов, таких как стили CSS, JavaScript-скрипты и изображения, обеспечивающая оптимальную производительность приложения. Также проведено сравнение методов и классов представлений, выявлены преимущества и недостатки каждого из подходов.

В результате исследовательской работы можно прийти к выводу, что выбор определенного подхода или решения зависит от конкретных требований проекта, его масштаба, сложности. Нет универсального решения, которое подходило бы для всех случаев, но понимание основных принципов и возможностей фреймворка Django позволяет разработчикам принимать обоснованные решения и создавать качественные веб-приложения.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Документация Django [Электронный ресурс]. − URL: https://djangodoc.ru/3.2/ (Дата обращения – 09.04.2024)
2. Фреймворк для веб-приложений на Python [Электронный ресурс]. URL: https://habr.com/ru/hubs/django/articles/ (Дата обращения – 10.04.2024)
3. Сайт администратора Django [Электронный ресурс]. URL: https://djangodoc.ru/3.2/ref/contrib/admin/ (Дата обращения – 12.04.2024)