

МИНОБРНАУКИ РФ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ДАГЕСТАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Факультет Информатики и Информационных Технологий

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
по дисциплине (модулю)
***«Средства проектирования и
сопровождение интернет приложений»***

**Кафедра Информационных Технологий
и Безопасности компьютерных систем**

Образовательная программа бакалавриата

09.03.02 Информационные системы и технологии

Направленность (профиль) программы:

Модуль профильной направленности

Форма обучения

Очная, заочная

Статус дисциплины:

входит в часть, формируемую участниками образовательных отношений;

Махачкала, 2021

**1. ПАСПОРТ
ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ
по дисциплине**

«Средства проектирования и сопровождение интернет приложений»

1.1. Основные сведения о дисциплине

Общая трудоемкость дисциплины составляет 3 зачетные единицы (108 академических часа).

Вид работы	Трудоемкость, академических часов		
	5 семестр	6 семестр	всего
Общая трудоёмкость		108	108
Контактная работа:			
Лекции (Л)		20	20
Практические занятия (ПЗ)			
Лабораторные занятия (ЛЗ)		34	34
Консультации			
Промежуточная аттестация (зачет, экзамен)		экзамен	экзамен
Самостоятельная работа:		54	54
- <i>написание реферата (Р);</i>		10	10
- <i>написание эссе (Э);</i>			
- <i>самостоятельное изучение разделов (перечислить);</i>		14	14
- <i>самоподготовка (проработка и повторение лекционного материала и материала учебников и учебных пособий);</i>		10	10
- <i>подготовка к практическим занятиям;</i>		10	10
- <i>подготовка к коллоквиумам;</i>		10	10
- <i>подготовка к рубежному контролю и т.п.)</i>			

1.2. Требования к результатам обучения по дисциплине, формы их контроля и виды оценочных средств

ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ

по дисциплине «Средства проектирования и сопровождение интернет приложений»

№ п/п	Контролируемые модули, разделы (темы) дисциплины	Код контролируемой компетенции (или её части)	Оценочные средства		Способ контроля
			наименование	№№ заданий	
1.	Основы проектирования веб-сайтов. Современные инструментальные средства веб-разработки. Размещение и сопровождение веб-сайтов. Web 2.0 программирование	ОПК-3, ОПК-9 ПК-7, ПК-8	Устный опрос,	1-9	Устно,
			письменный опрос,	1	Письменно
			лабораторная работа	1	Компьютерная реализация
2.	Основы HTML.	ОПК-3, ОПК-9 ПК-7, ПК-8	Устный опрос,	10-15	Компьютерное тестирование
			письменный опрос,	1	устно
			лабораторная работа	2	Компьютерное тестирование
3.	Основы CSS Стилевые таблицы	ОПК-3, ОПК-9	Устный опрос,	10-15	Компьютерное тестирование

		ПК-7, ПК-8	письменный опрос,	1	устно
			лабораторная работа	2	Компьютерное

					тестирование
4.	Ведение в Python: Синтаксис. Структуры данных. Функции. Классы Общая характеристика языка Python		Устный опрос,	16-	Устно,
			письменный опрос,	1	Письменно
			лабораторная работа	3-4	Компьютерная реализация
5.	Использование шаблона MVC в Django.		Устный опрос,	16-20	Устно,
			письменный опрос,	2	Письменно
			лабораторная работа	5	Компьютерная реализация
6.	Структура веб-проекта Django		Устный опрос,	16-20	Устно,
			письменный опрос,	2	Письменно
			лабораторная работа	5	Компьютерная реализация
7.	Наследование шаблонов. Макет на основе мастер-страницы		Устный опрос,	21-25	Устно,
			письменный опрос,	3	Письменно
			лабораторная работа	6	Компьютерная реализация
8.	Веб-формы в Django. Валидация. JavaScript		Устный опрос,	26-33	Устно,
			письменный опрос,		Письменно
			лабораторная работа	7	Компьютерная реализация
9.	Административный раздел Django. Аутентификация и авторизация в Django		Устный опрос,	34-48	Устно,
			письменный опрос,	4	Письменно
			лабораторная работа	8	Компьютерная реализация
10.	Использование баз данных в Django		Устный опрос,	34-48	Устно,
			письменный опрос,	4	Письменно
			лабораторная работа	9	Компьютерная реализация

**2. КОНТРОЛЬНЫЕ ЗАДАНИЯ И ИНЫЕ МАТЕРИАЛЫ ОЦЕНКИ
знаний, умений, навыков и (или) опыта деятельности,
характеризующие этапы формирования компетенций в процессе освоения
дисциплины**

«Средства проектирования и сопровождение интернет приложений»

Вопросы для коллоквиумов, собеседования

1. Где можно зарегистрировать доменное имя для сайта?
2. Какие критерии используются для выбора хостинга?
3. Для чего необходим бесплатный тестовый период у хостинга?
4. Опишите основные отличия HTML документов от обычных текстовых документов.
5. Расскажите об отличиях одиночных тегов от контейнерных тегов.
6. Перечислите наиболее широко используемые блочные теги.
7. Опишите основные типы селекторов, используемых в стилевых правилах CSS
8. Расскажите о различных способах подключения стилей к веб-странице.
9. Перечислите наиболее широко используемые стилевые свойства, управляющие позиционированием и компоновкой элементов.
10. Какие команды надо выполнить в среде Visual Studio для создания и сохранения проекта программы на языке Python?
11. Как преобразовать вводимые пользователем программы символы в числовые

- значения?
12. Как округлить с заданной точностью результат вычислений?
 13. Как вывести результаты вычислений на экран?
 14. Какие команды надо выполнить в среде Visual Studio для создания веб-проекта Django?
 15. Каким образом можно запустить веб-приложение на выполнение?
 16. Какие изменения необходимо внести, чтобы сделать русскоязычный интерфейс в веб-приложении?
 17. Каким образом можно добавить логотип в свой веб-сайт?
 18. Что необходимо сделать, чтобы осуществлялся переход на Главную страницу при нажатии на логотип?
 19. Какие расширения графических файлов могут быть у изображений, размещенных на веб-сайте.
 20. Какие действия следует выполнить, чтобы добавить в веб-проект Django страницу сайта, содержащую форму?
 21. Каким образом описывается состав полей формы в веб-проекте Django?
 22. Как элементы формы в Django включаются в состав шаблона веб-страницы?
 23. Какие действия должен выполнить метод, обрабатывающий запрос к странице с формой?
 24. Как получить в методе, обрабатывающем отправку формы, доступ к значениям полей формы, успешно прошедшей валидацию?
 25. Как передать переменную формы с данными в шаблон страницы?
 26. Какие команды надо выполнить в среде Visual Studio для создания администратора веб-сайта?
 27. Какие действия необходимо выполнить для входа в административный раздел сайта?
 28. Что необходимо сделать, чтобы добавить страницу «Регистрация» на сайт?
 29. Каким образом можно зарегистрировать тестового пользователя?
 30. Каким образом в веб-проекте Django может быть реализована авторизация пользователей.
 31. Какие команды надо выполнить в среде Visual Studio для создания таблиц базы данных по моделям?
 32. В какой файл необходимо добавить новую модель?
 33. Какие существуют типы полей модели?
 34. Какова структура типичной модели Django?
 35. Для чего используются метаданные модели?
 36. Где можно добавлять данные статей блога?
 37. Какие команды необходимо выполнить в среде Visual Studio для добавления формы комментария в веб-проект?
 38. Что необходимо сделать для отображения всех комментариев к статье блога?
 39. Какой метод предназначен для переадресации на конкретную веб-страницу?
 40. Каким образом передают данные полученные из модели в шаблон веб-страницы?
 41. Каким образом можно реализовать добавление комментариев к постам только авторизованными пользователями?
 42. Какие команды надо выполнить в среде Visual Studio для добавления в проект новой веб-страницы?
 43. В какой файл необходимо добавить URL для страницы поста?
 44. Какие существуют способы размещения списка статей блога на веб-странице?
 45. Какова должна быть структура типичной статьи блога?
 46. Что необходимо сделать, чтобы отобразить конкретную статью блога на отдельной веб-странице?
 47. Каким образом получить все статьи блога из базы данных?
 48. Какие настройки необходимо выполнить, чтобы графические файлы загружались на

- сайт в папку media через форму?
49. Какое поле требуется добавить в модель блога, чтобы статья блога содержала еще и изображение?
 50. Опишите действия, которые необходимо выполнить для добавления в проект формы для ввода статьи блога.
 51. Что необходимо сделать, чтобы отобразить картинку на веб-странице до текста статьи блога?
 52. Что необходимо сделать, чтобы новые статьи блога на веб-странице сайта мог добавлять только администратор?
 53. Какого формата видео-файлы должны быть включены в источники тега <video>, чтобы он воспроизводился в разных браузерах?

Критерии оценки:

- оценка **«отлично»** выставляется студенту, если показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений;
- оценка **«хорошо»** выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности
- оценка **«удовлетворительно»** выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом он владеет основными разделами учебной программы, необходимыми для дальнейшего обучения и может применять полученные знания по образцу в стандартной ситуации;
- оценка **«неудовлетворительно»** выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом он владеет основными разделами учебной программы, необходимыми для дальнейшего обучения и может применять полученные знания по образцу в стандартной ситуации;

Темы эссе (рефератов, докладов, сообщений)

1. Введение в средства проектирования и сопровождение интернет приложений.
2. Проектирование интерфейсов пользовательского интерфейса для веб-приложений.
3. Разработка архитектуры веб-приложений и выбор технологий.
4. Создание баз данных и управление данными в веб-приложениях.
5. Разработка и тестирование API для веб-приложений.
6. Интеграция веб-приложений с другими системами.
7. Оптимизация производительности веб-приложений.
8. Разработка систем безопасности для веб-приложений.
9. Автоматизированное тестирование веб-приложений.
10. Мониторинг и сопровождение веб-приложений в производственной среде.

Критерии оценки:

- оценка **«отлично»** выставляется студенту, если он выполнил работу самостоятельно, сделал выводы, объяснил результаты;
- оценка **«хорошо»** выставляется студенту, если он выполнил работу с помощью преподавателя или других студентов, сделал выводы, объяснил результаты;

- оценка «удовлетворительно» выставляется студенту, если он выполнил работу с помощью преподавателя или других студентов, но не сделал выводы самостоятельно;
- оценка «неудовлетворительно» выставляется студенту, если он не выполнил работу или выполнил частично.
- оценка «зачтено» выставляется студенту, если он выполнил работу.;
- оценка «не зачтено» выставляется студенту, если он не выполнил работу или выполнил частично.

Лабораторные работы

Лабораторная работа 1. Выбор хостинга, доменного имени

1. необходимо:
 - а. Скопировать адрес сайта REG.RU и вставить его в адресную строку браузера, чтобы загрузить сайт REG.RU
 - б. Ниже на главной странице сайта (см. рисунок 1) выполнить следующие действия:
 - установить флажок зоны **RU** (см. рисунок 2)
 - в поле для подбора домена ввести желаемое название домена 2 уровня и нажать кнопку **Подобрать** (см. рисунок 2)
 - посмотрите загруженные результаты проверки доменного имени и стоимость выбранного доменного имени (см. рисунок 3).

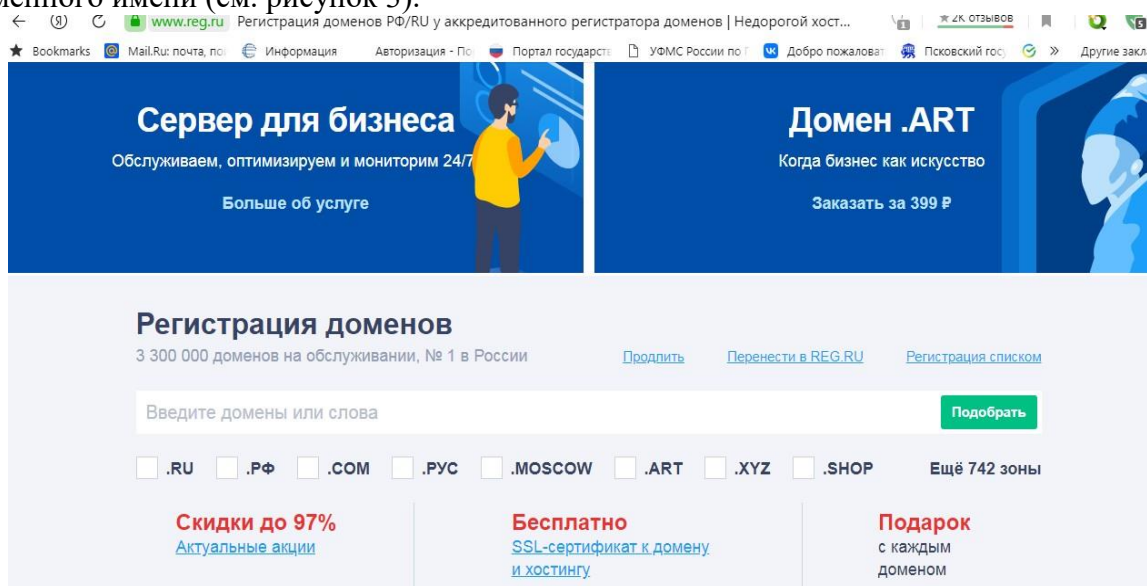


Рисунок 1 – Подбор доменного имени на главной странице сайта REG.RU

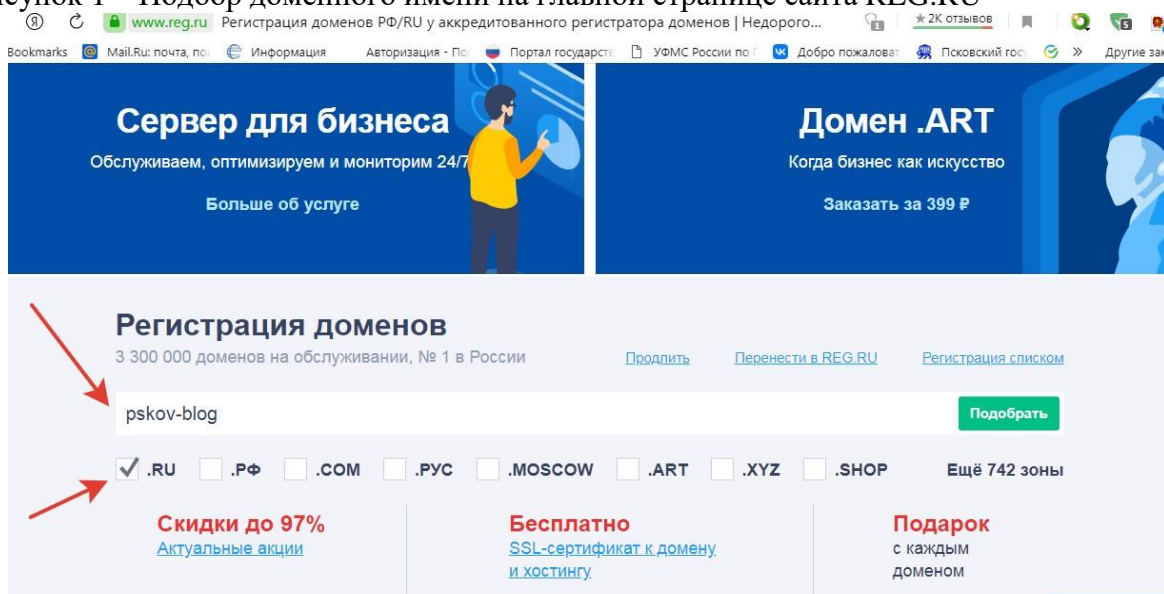


Рисунок 2 – Подбор доменного имени на сайте REG.RU

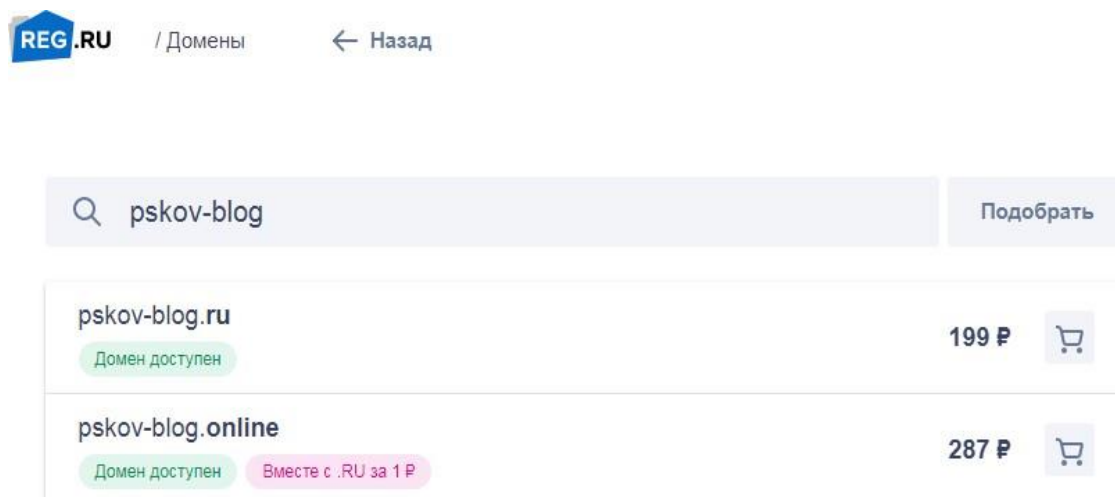


Рисунок 3 – Результаты проверки доменного имени и стоимость выбранного доменного имени на сайте REG.RU

2. На основе данных в подразделе **Теория для практического занятия** необходимо:

- а. Перейти на сайте REG.RU (или ввести в адресной строке браузера адрес сайта REG.RU из **Таблицы 1 задания 1**) на страницу «Хостинг» (см. рисунок 4). На странице выполнить следующие действия:
 - посмотреть тарифы на вкладках отображения тарифов **Эконом** и **Скоростные** найти подходящий по критерию отбора в задании 1 тариф, т.е. тариф с поддержкой Python и с минимальной стоимостью (см. рисунок 5) – это название понадобится для заполнения **Таблицы 1 п.3 задания 1**

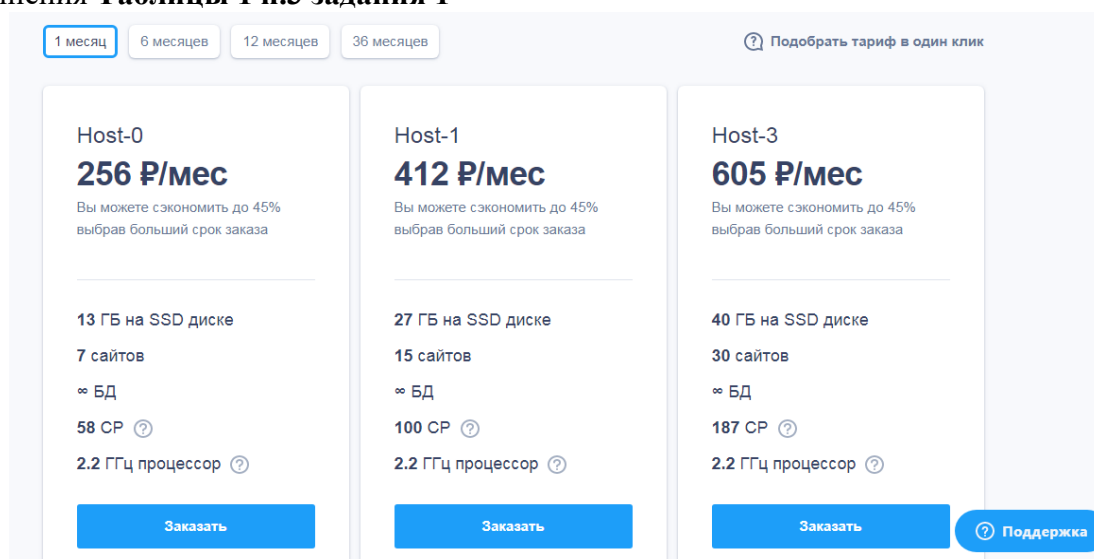


Рисунок 4 – Веб-страница **Хостинг** на сайте REG.RU

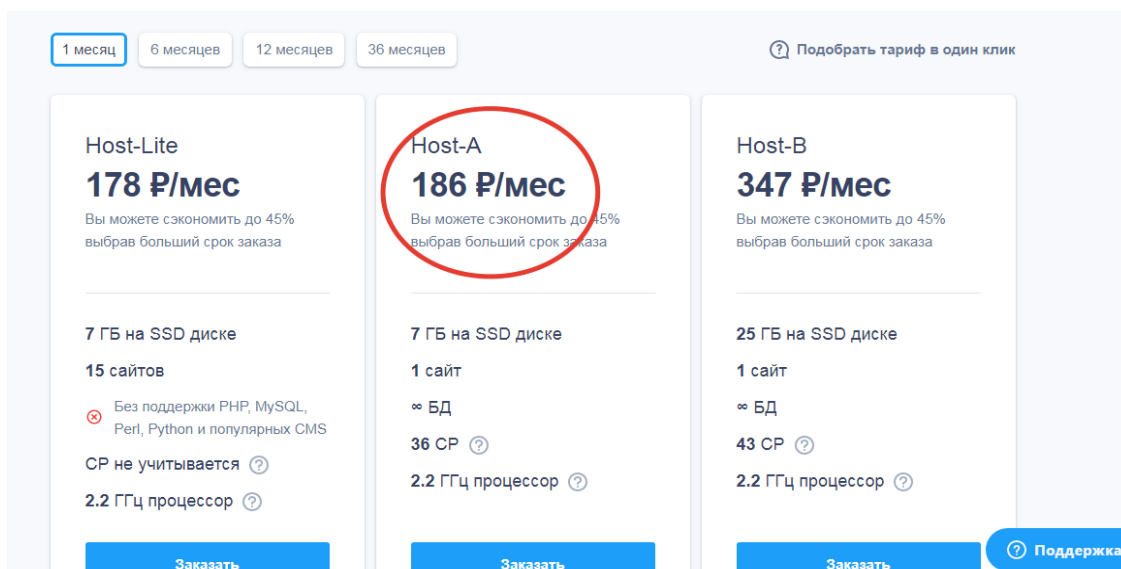


Рисунок 5 – Страница с выбранным тарифом

в. Ниже на этой же странице расположена информация о бесплатном тестовом периоде для хостинга в днях – 14 дней – эта информация также понадобится для заполнения **Таблицы 1 п. 3 задания 1** (см. рисунок 6)

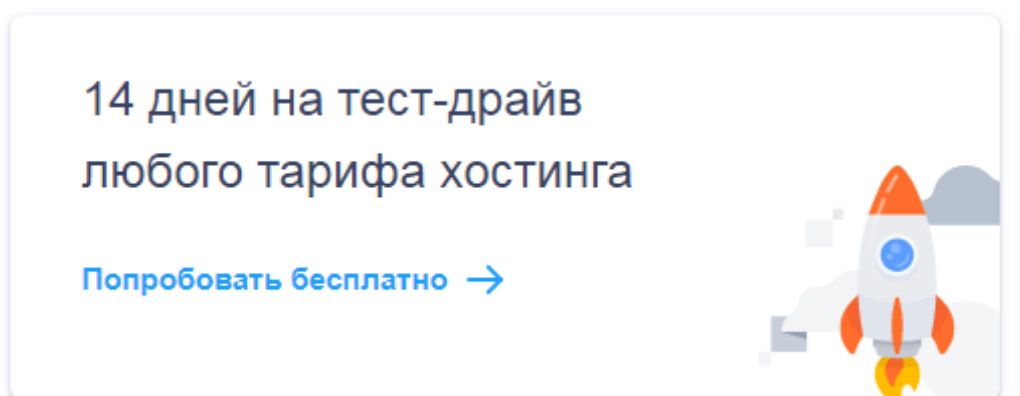


Рисунок 6 – Данные о бесплатном тестовом периоде для хостинга в днях

с. Ввести в адресной строке браузера адрес следующего сайта хостинговой компании из **Таблицы 1 задания 1**. На странице выполнить действия по выбору тарифа и определения бесплатного тестового периода для хостинга в днях по аналогии с предыдущими пунктами (а и в). **Повторить для всех сайтов хостинг-компаний из Таблицы 1 задания 1**

3. На основе данных п.2 заполнить **Таблицу 1** задания 1:

Таблица 1

№	Хостинговая компания	Название тарифа с поддержкой Python	Бесплатный тестовый период
1	REG.RU http://hosting.reg.ru		
2	SpaceWeb http://sweb.ru		
3	Beget https://beget.com		

Лабораторная работа 2. Создание веб-страниц, содержащих гиперссылки и изображения

1. Для выполнения первого шага следует запустить Проводник Windows, выбрать любой раздел для папки с веб-страницами (например, **D:\TMP**) и по правой кнопке мыши выбрать в контекстном меню команду **Создать – Папку**. Созданную папку переименовать, например, в **www**.

Из теории в подразделе **Теория для практического занятия** следует, что первоначально необходимо:

- а. Запустить на своем компьютере **Visual Studio**.
- б. Далее в меню **Файл** выбрать команду **Создать**, далее выбрать **Файл**. В появившемся окне выбрать в списке «HTML-страница» и нажать кнопку «Создать». В результате будет создан шаблон стандартной HTML-страницы (см. рисунок 1).

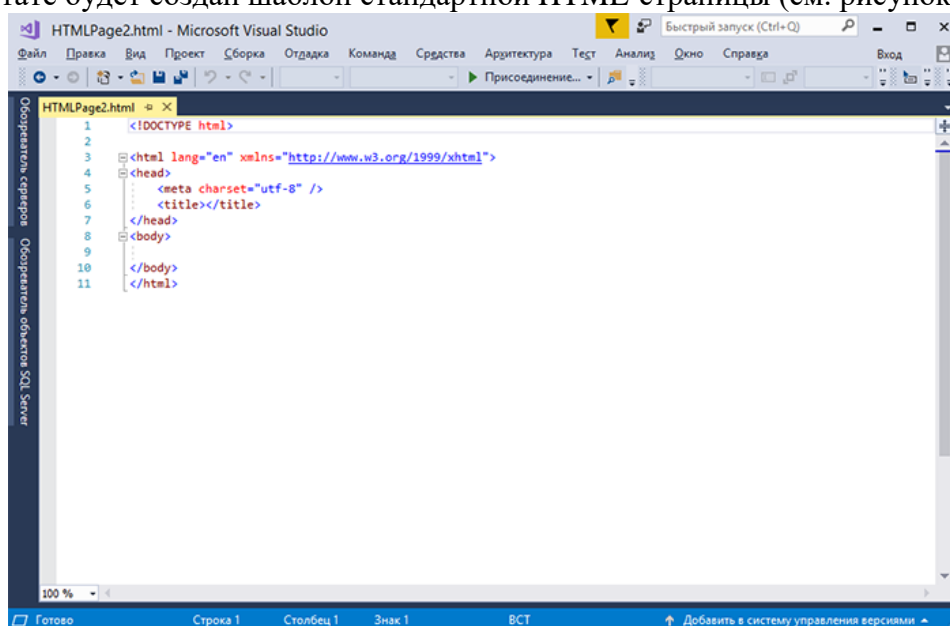


Рисунок 1 – Окно с созданным HTML-документом

- с. Командой меню **Файл** «Сохранить <имя файла> как...» сохранить созданный файл под именем **about.html** в папке для файлов сайта (см. рисунок 2).

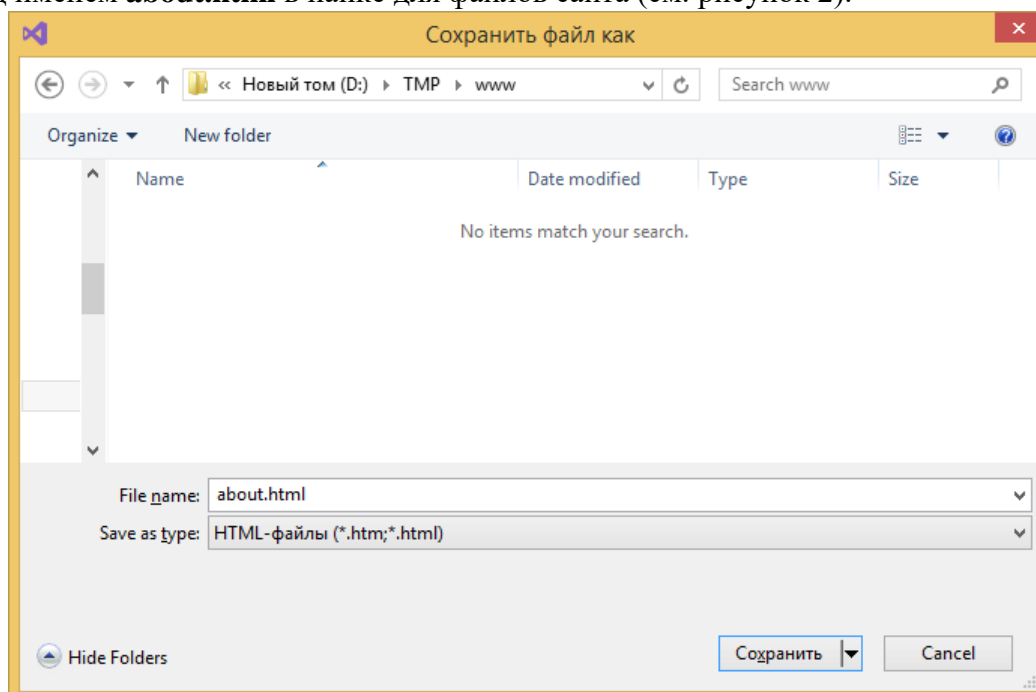


Рисунок 2 – Сохранение HTML-документа

2. Подобранные для страниц сайта текстовые материалы можно сохранить в отдельных текстовых файлах или сразу разместить в секции body HTML-файлов проекта. Графические файлы следует скопировать в папку файлов веб-сайта.

3. Из теории в подразделе **Теория для практического занятия** следует, что для редактирования файлов в папке веб-сайта необходимо:

- а. В меню **Файл** выбрать команду Открыть далее выбрать «Файл». В появившемся окне выбрать файл и нажать кнопку «Открыть».
- б. Выполнить необходимые изменения.
- с. Командой меню **Файл** «Сохранить» сохранить изменения. На рисунке 5 представлен файл с добавленной разметкой.

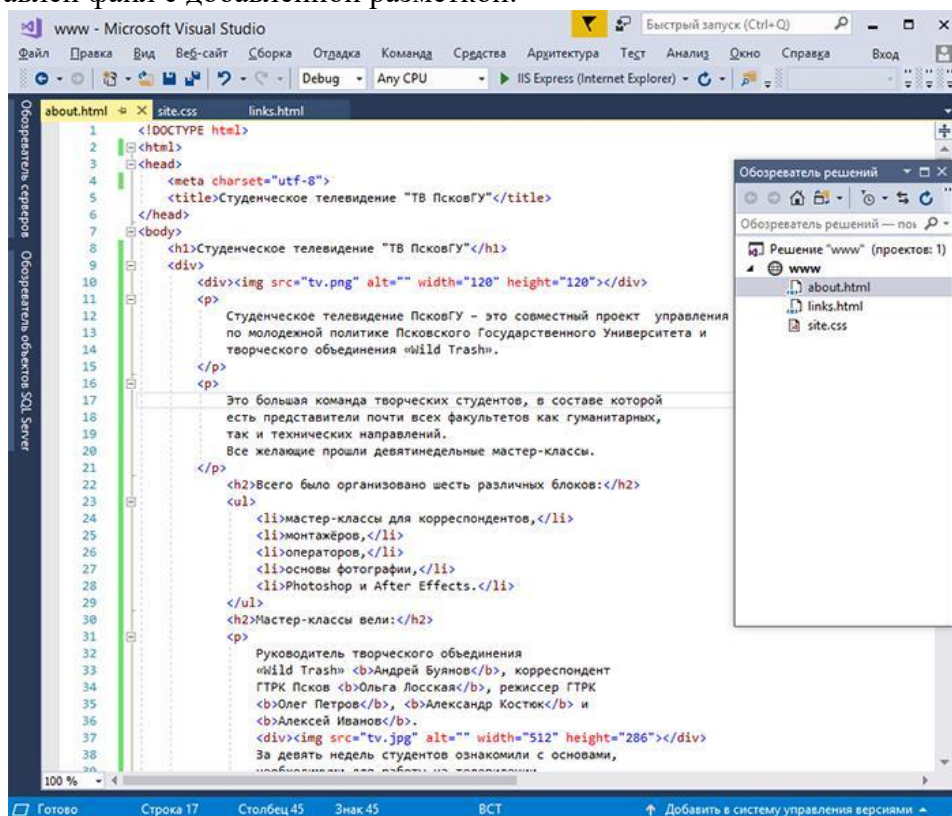


Рисунок 5 – Просмотр разметки HTML-документа

- d. Для просмотра отображения созданного документа в браузере для сохраненного файла следует зайти в папку сайта через Проводник Windows и выполнить для файла команду контекстного меню «Открыть с помощью», выбрав в списке программ установленный в системе браузер. На рисунке 6 представлен результат просмотра в браузере.

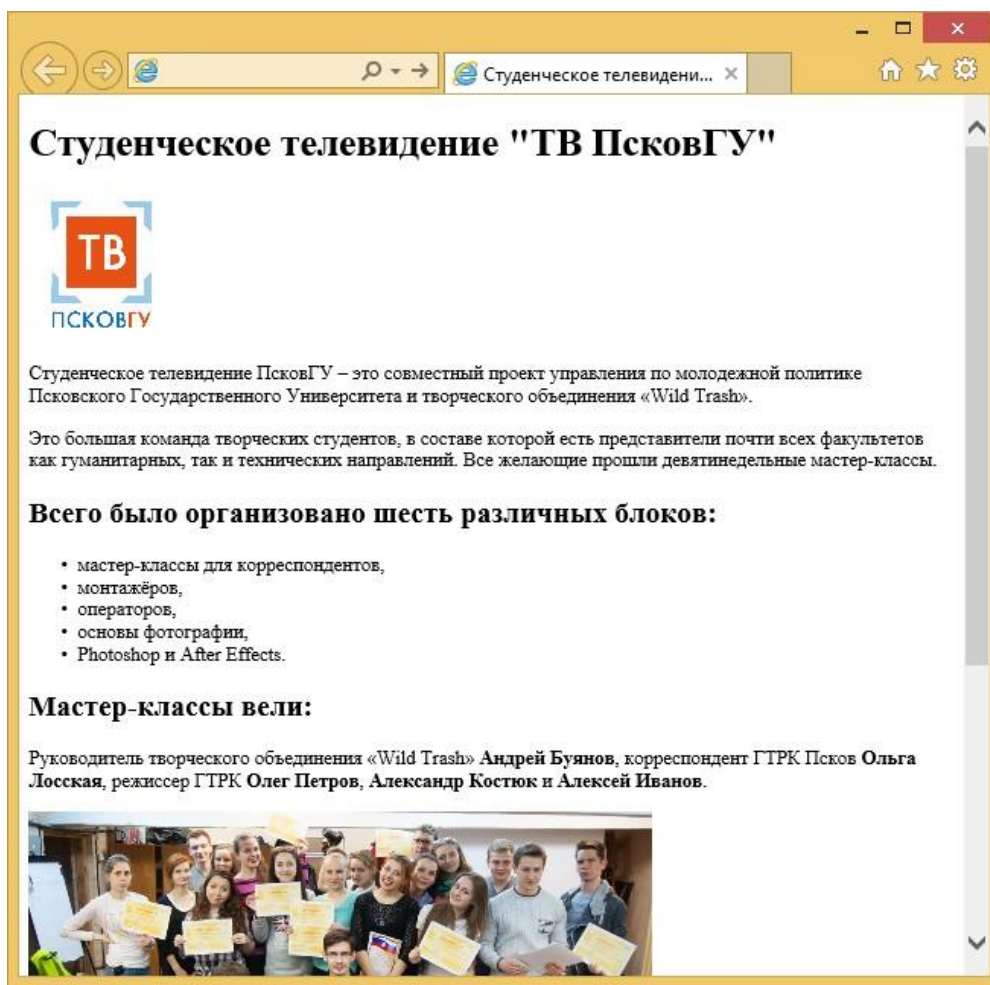
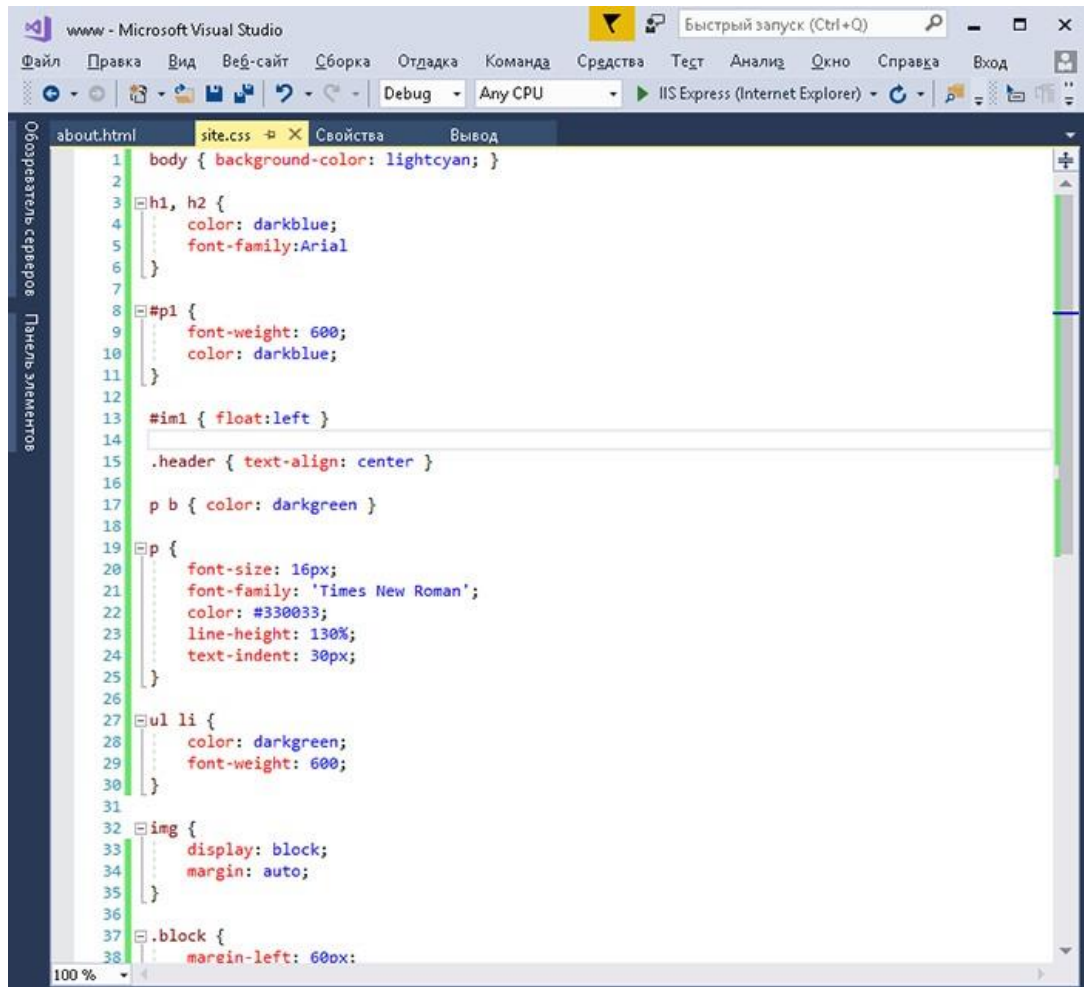


Рисунок 6 – Просмотр документа в браузере

5. Редактирования файла таблицы стилей выполняется аналогично редактированию файлов с HTML-разметкой

а. На рисунке 7 представлен файл таблицы стилей с добавленными стилевыми правилами.



```
1 body { background-color: lightcyan; }
2
3 h1, h2 {
4     color: darkblue;
5     font-family:Arial
6 }
7
8 #p1 {
9     font-weight: 600;
10    color: darkblue;
11 }
12
13 #im1 { float:left }
14
15 .header { text-align: center }
16
17 p b { color: darkgreen }
18
19 p {
20     font-size: 16px;
21     font-family: 'Times New Roman';
22     color: #330033;
23     line-height: 130%;
24     text-indent: 30px;
25 }
26
27 ul li {
28     color: darkgreen;
29     font-weight: 600;
30 }
31
32 img {
33     display: block;
34     margin: auto;
35 }
36
37 .block {
38     margin-left: 60px;
```

Рисунок 7 – Таблица стилей

в. На рисунке 8 представлен HTML-файл с подключенной таблицей стилей и добавленными в теги атрибутами **id** и **class**.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>Студенческое телевидение "ТВ ПсковГУ"</title>
6 <link rel="stylesheet" href="site.css" type="text/css">
7 </head>
8 <body>
9 <h1 class="header">Студенческое телевидение "ТВ ПсковГУ"</h1>
10 <div class="block">
11 <div></div>
12 <p id="p1">
13 Студенческое телевидение ПсковГУ – это совместный проект управления
14 по молодежной политике Псковского Государственного Университета и
15 творческого объединения «Wild Trash».
16 </p>
17 <p>
18 Это большая команда творческих студентов, в составе которой
19 есть представители почти всех факультетов как гуманитарных,
20 так и технических направлений.
21 Все желающие прошли девятидневные мастер-классы.
22 </p>
23 <h2>Всего было организовано шесть различных блоков:</h2>
24 <ul>
25 <li>мастер-классы для корреспондентов,</li>
26 <li>монтажёров,</li>
27 <li>операторов,</li>
28 <li>основы фотографии,</li>
29 <li>Photoshop и After Effects.</li>
30 </ul>
31 <h2>Мастер-классы вели:</h2>
32 <p>
33 Руководитель творческого объединения
34 «Wild Trash» <b>Андрей Буянов</b>, корреспондент
35 ГТРК Псков <b>Ольга Лосская</b>, режиссер ГТРК
36 <b>Олег Петров</b>, <b>Александр Костюк</b> и
37 <b>Алексей Иванов</b>.
38 <div><ime src="tv.foe" alt="" width="512" height="286"></div>
```

Рисунок 8 – HTML-документ со стилевым оформлением с. На рисунке 9 представлен результат просмотра в браузере страницы с добавленным стилевым оформлением

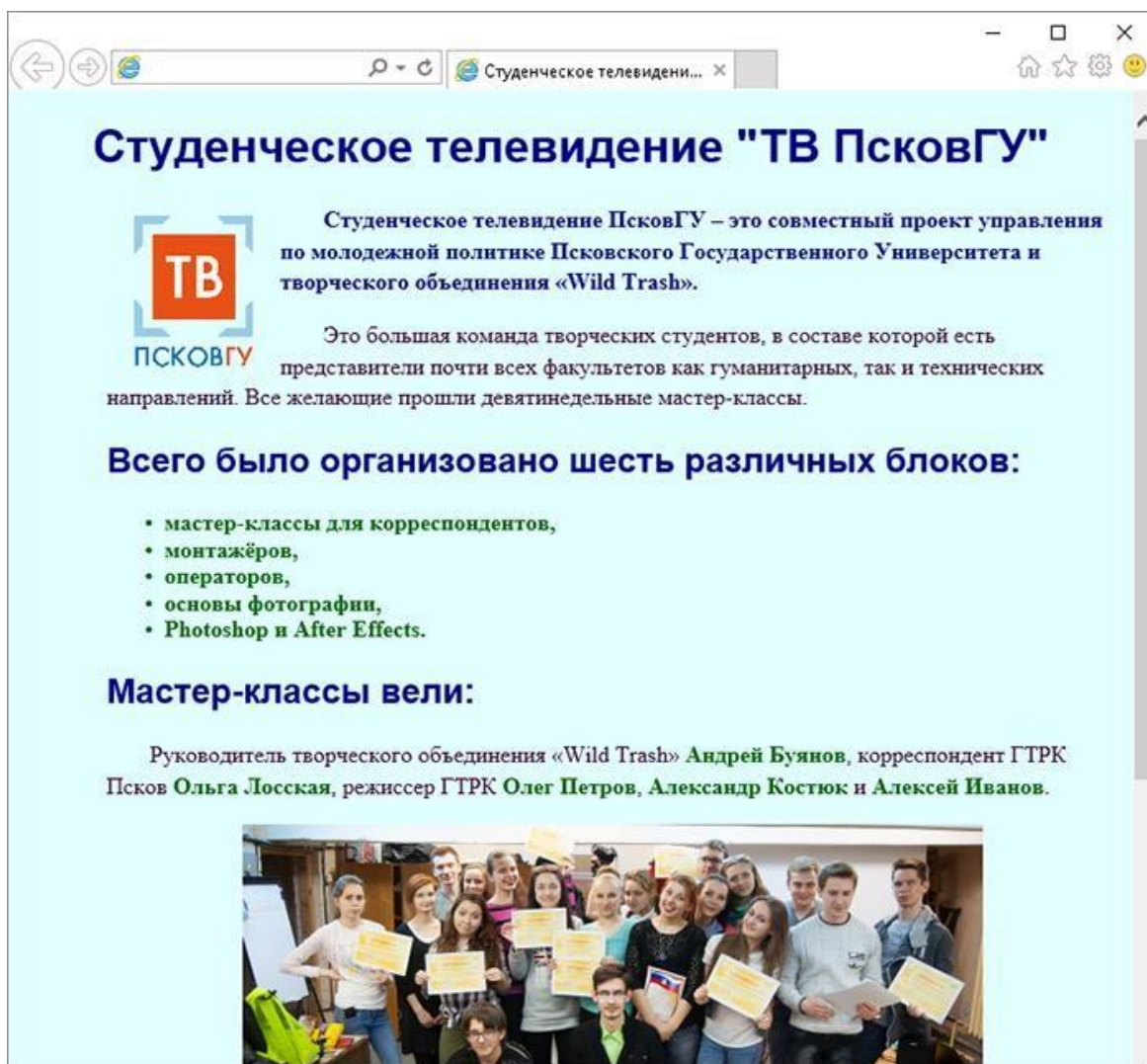


Рисунок 9 – Просмотр документа со стиливым оформлением в браузере

6. Для проверки успешности выполнения работы следует открыть и просмотреть в браузере созданные веб-страницы. Убедиться в успешном отображении выполненной разметки и стилового оформления и в успешном переходе по ссылкам между созданными страницами и на ресурсы в интернете.

Лабораторная работа 3. Решение задачи на операции с числами и строками на Python

В примере решается задача по вычислению и выводу длины окружности и площади круга на основе ввода длины радиуса пользователем программы.

1. Из теории в подразделе **Теория для практического занятия** следует, что первоначально необходимо:
 - a. Запустить на своем компьютере **Visual Studio**.
 - b. Перейти в панель "**Создание проекта**" (см. рисунок 1):
 - в списке языков вверху выбрать **Python**
 - в списке типов проектов выбрать **Приложение Python**
 - нажать кнопку **Далее**
 - в открывшейся панели выбрать путь к папке проекта и нажать кнопку "**Создать**"

Рисунок 1 – Окно «Создание проекта»

2. Из теории в подразделе **Теория для практического занятия** следует, что для ввода текста программы потребуется:

а. Открыть для редактирования файл PythonApplication1.py через «Обозреватель решений» (см. рисунок 2).

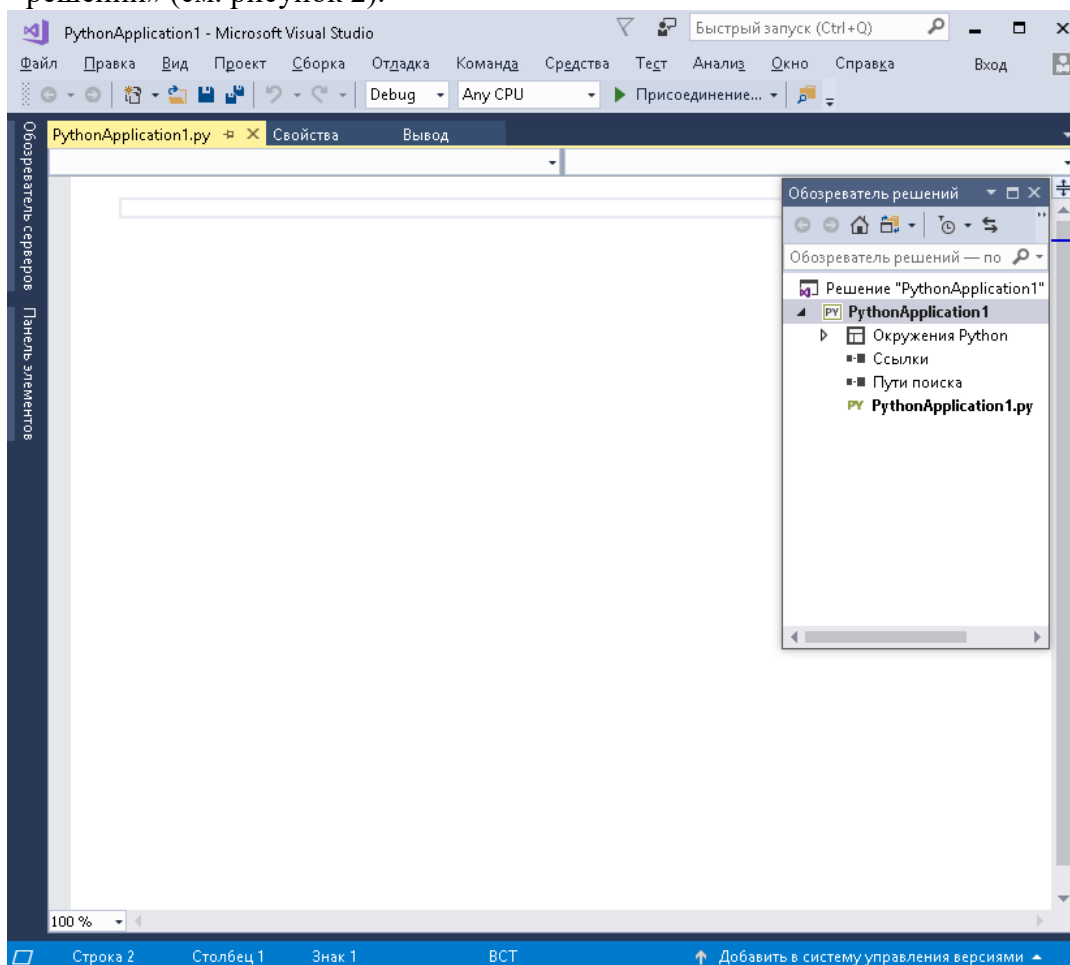


Рисунок 2 – Открытый для редактирования файл PythonApplication1.py

б. Ввести текст программы в открытое для редактирования окно (см. рисунок 3).

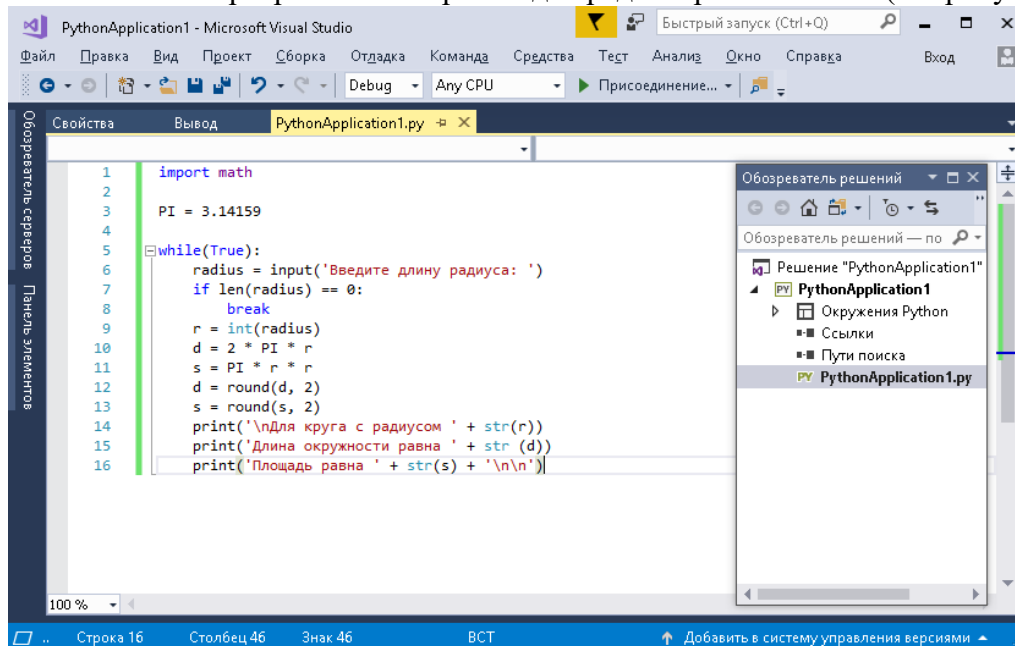


Рисунок 3 – Набранный текст программы

Комментарий к тексту программы из примера

В строке 1 подключается пакет с математическими функциями math.

В строке 3 задается значение для константы π . Имена констант в Python традиционно указываются в верхнем регистре.

В строке 5 задается бесконечный цикл `while`, выйти из которого можно, используя оператор `break`.

В строке 6 в переменную `radius` записывается введенный пользователем текст.

В строке 7 проверяется длина строки, введенной пользователем, и если она равна нулю, то в строке 8 цикл прерывается оператором `break` и программа завершается.

В строке 9 введенная пользователем строка преобразуется в целочисленное значение функцией `int()`.

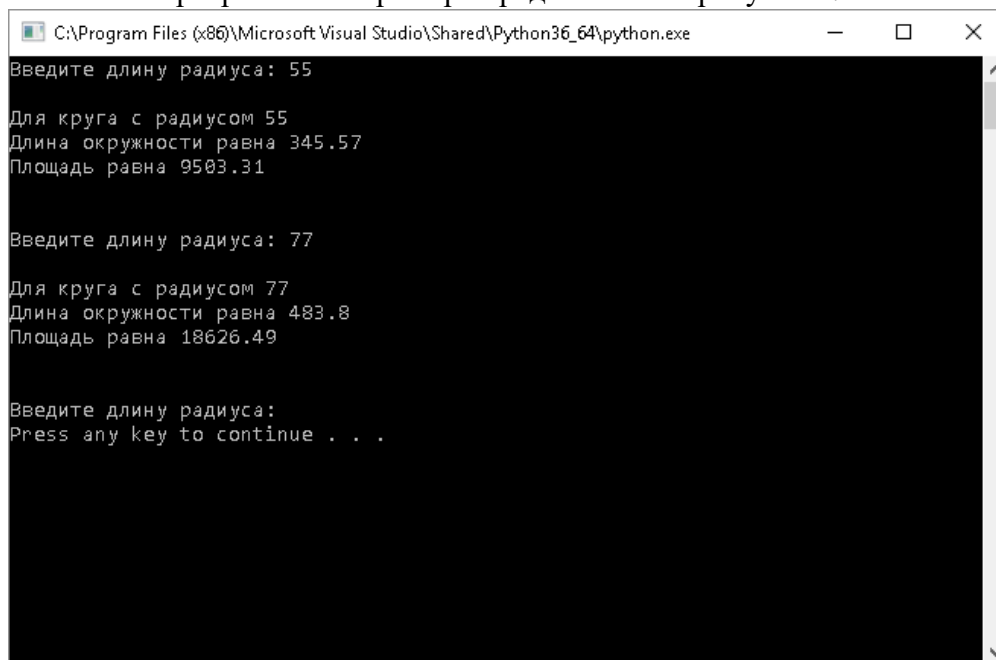
В строках 10 и 11 вычисляются длина окружности и площадь круга.

В строках 12 и 13 вычисленные значения округляются до двух знаков после запятой.

В строках 14, 15, 16 формируются строки, содержащие результаты вычислений, и передаются функции `print()` для вывода на экран. Последовательность символов «`\n`» представляет собой специальную последовательность, задающую перевод строки при выводе.

3. Из теории в подразделе **Теория для практического занятия** следует, что для запуска набранной программы следует:

- a. В меню **Проект** выполнить команду "Запуск без отладки". Результат выполнения программы из примера представлен на рисунке 4.



```
C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python36_64\python.exe
Введите длину радиуса: 55
Для круга с радиусом 55
Длина окружности равна 345.57
Площадь равна 9503.31

Введите длину радиуса: 77
Для круга с радиусом 77
Длина окружности равна 483.8
Площадь равна 18626.49

Введите длину радиуса:
Press any key to continue . . .
```

Рисунок 4 – Выполнение программы

- b. При выявлении ошибок следует их исправить и повторно запустить программу на выполнение.

Лабораторная работа 4. Создание веб-проекта Django

a. Запустить на своем компьютере **Visual Studio**. В появившемся окне:

- выбрать **Создание проекта** → **Веб-проект Django** (рисунок 1).
- далее в окне «**Настроить новый проект**» задать **Имя проекта** и **Расположение** и нажать кнопку **Создать** (рисунок 2).

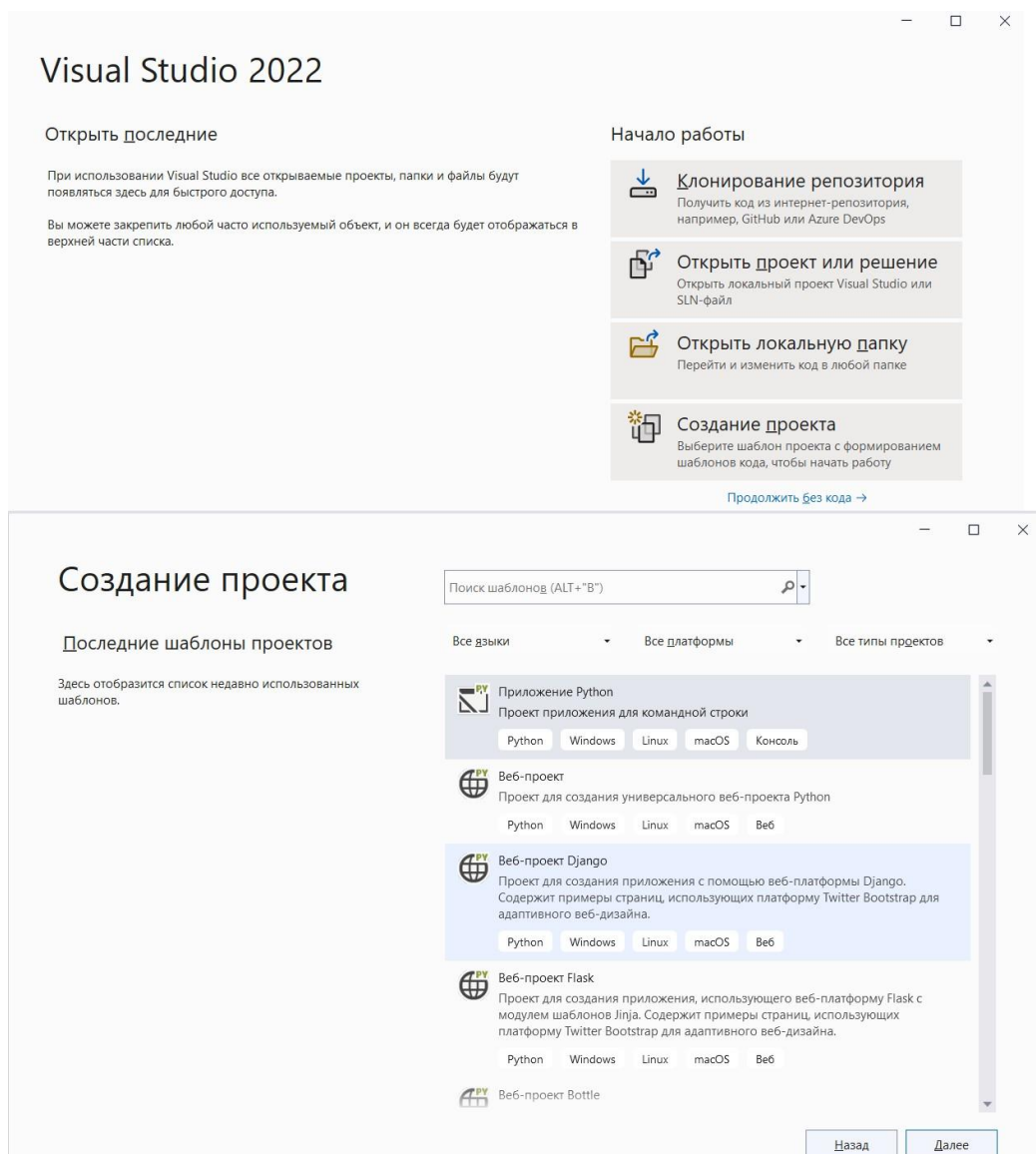


Рисунок 1 – Создание Веб-проекта Django

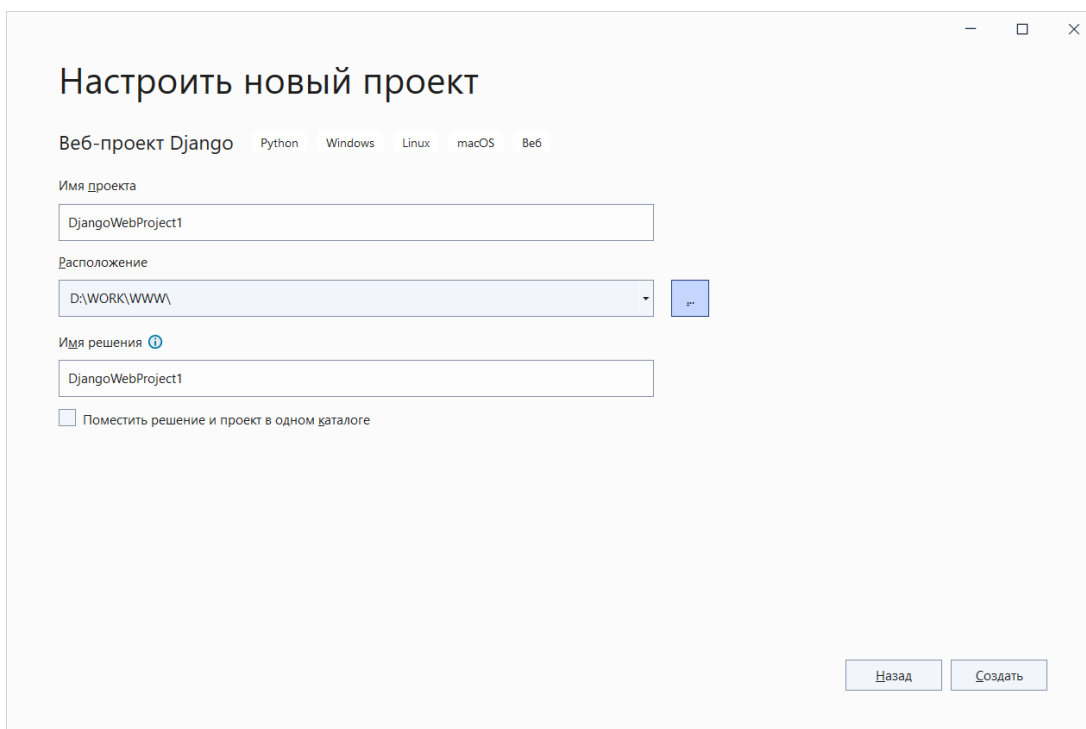


Рисунок 2 – Настроить новый проект

с. Далее установить внешние пакеты в проект: для этого в "Обозревателе решений" по клику правой кнопкой мыши по строке "Окружения Python" выбрать команду "Добавить среду", затем в появившейся панели выберите в поле "Базовый интерпретатор" – Python 3.9 (или Python 3.7 для Windows 7) и нажать кнопку Создать (рисунок 3).

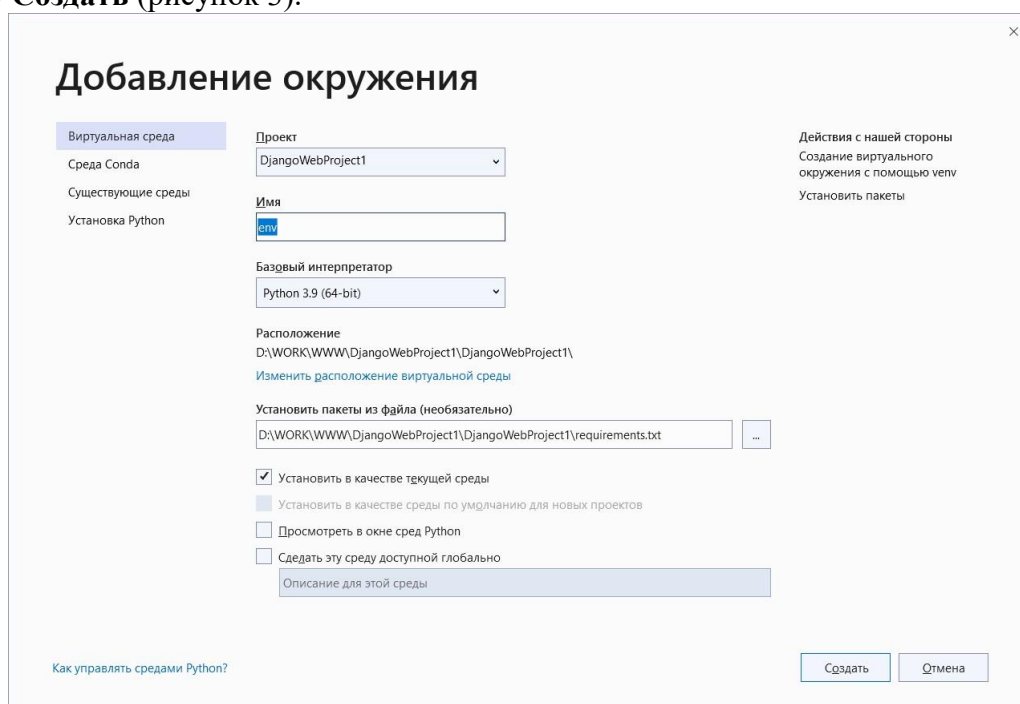


Рисунок 3 – Добавление виртуального окружения

2. Из теории в подразделе **Теория для практического занятия** следует, что при первом запуске веб-приложения можно в настройке способа отладочного запуска **выбрать браузер** для запуска приложения – например, Internet Explorer (см. рисунок 4).

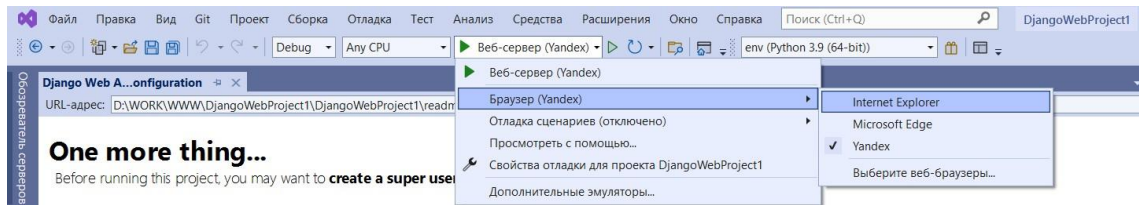


Рисунок 4 – Выбор браузера

Затем запустить веб-приложение, нажав кнопку **Веб-сервер (Internet Explorer)**.

Для завершения работы веб-приложения требуется нажать кнопку закрытия окна браузера, а затем на панели инструментов Visual Studio нажать кнопку **Остановить отладку** (см. рисунок 5).

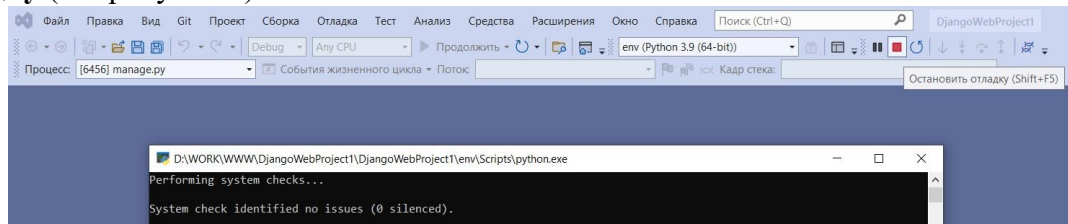


Рисунок 5 – Остановить отладку

3. На основе материалов конспекта лекции 5, презентации к лекции 5 в окне «Обозреватель решений» посмотреть структуру веб-проекта Django. Открыть и посмотреть содержание файлов:

1.

- модели models.py;
- контроллера views.py;
- представлений в папке \app\templates\app – about.html, contact.html, index.html;
- Мастер-страницы – layout.html;
- urls.py;
- settings.py

4. Из теории в подразделе **Теория для практического занятия** следует, что для того, чтобы заменить меню, заголовки и тексты стандартных макетов страниц шаблона сайта на соответствующие русскоязычные названия и тексты, необходимо:

a. В файле **settings.py** заменить:

```
LANGUAGE_CODE = 'en-us'
```

на

```
LANGUAGE_CODE = 'ru-ru'
```

b. Затем необходимо каждый html-файл (about.html, contact.html, index.html, layout.html, login.html, loginpartial.html) из папки веб-проекта **\app\templates\app** сохранить в кодировке **UTF-8**:

- выбрать в меню **Файл** команду **Сохранить <имя файла> как...**
- в диалоговом окне выбрать в списке кнопки **Сохранить** команду **Сохранить с кодировкой...**
- подтвердить сохранение с заменой существующего файла (нажать кнопку **Да**)
- в диалоговом окне **Дополнительные параметры сохранения** выбрать кодировку **Юникод (UTF-8)** и нажать **ОК**.
- перезагрузить сохраненный файл.

c. В файле **views.py** названия заголовков страниц, сообщений заменить в соответствующих методах действия контроллера на русскоязычные:

- `'title': 'Главная',`
- `'title': 'Контакты',`
- `'message': 'Страница с нашими контактами.'`
- `'title': 'О нас',`
- `'message': 'Сведения о нас.'`

d. В html-файлах необходимо заменить англоязычные тексты (названия) на русскоязычные в соответствии с выбранной темой для веб-сайта. Например, код файла **contact.html** может быть заменен на следующий:

```
{% extends "app/layout.html" %}

{% block content %}

<h2>{{ title }}.</h2>
<h3>{{ message }}</h3>

<address>
    Фирма<br />
    Улица, дом<br />
    <abbr title="телефон">Р:</abbr>
    425.555.0100
</address>

<address>
    <strong>Техническая поддержка:</strong> <a
href="mailto:Support@example.com">Support@example.com</a><br />
    <strong>Отдел маркетинга:</strong> <a
href="mailto:Marketing@example.com">Marketing@example.com</a>
</address>

{% endblock %}
```

5. Из теории в подразделе **Теория для практического занятия** следует, что для того, чтобы поместить найденный графический логотип сайта в верхней области страниц и, чтобы клик по логотипу приводил к переходу на главную страницу сайта, необходимо:

a. Найти у себя на компьютере или в сети интернет или самостоятельно создать в графическом редакторе, подходящий для логотипа вашего веб-сайта файл с расширением jpg или png. Например, файл **logo.jpg**.

b. Скопировать этот файл в свой веб-проект в папку **app\static\app\content**

c. В файле **templates\app\layout.html** заменить:

```
<a href="/" class="navbar-brand">Application name</a>
```

На следующий код:

```
<a href="/" class="navbar-brand">  </a>
```

d. В файле стилей **app\static\app\content\site.css** для тега **body** подобрать значение параметра **padding-top**, которое обеспечит отсутствие перекрытия шапки сайта с содержанием страниц.

Например, было изначально:

```
padding-top: 50px;
```

требуется изменить на:

```
padding-top: 80px;
```

6. Для проверки правильности отображения веб-страниц в браузере необходимо:

- запустить веб-приложение на выполнение;
- посмотреть корректно ли отображается русскоязычный контент веб-страниц в браузере;
- проверить переход по логотипу на главную страницу.

Лабораторная работа 5. Создание формы

a. Запустить на своем компьютере **Visual Studio**.

b. В стартовом окне выбрать "**Открыть проект или решение**" или в меню **Файл** выбрать команду **Открыть**, далее выбрать **Решение или проект...** В появившемся окне "**Открыть проект**" перейти в папку ранее созданного проекта, выбрать там файл с расширением .sln и нажать кнопку "Открыть".

с. Далее в окне «Обозреватель решений» выбрать и открыть для редактирования файл `app/forms.py`. Этот файл следует дополнить описанием класса создаваемой формы. На рисунке 1 представлен текст класса модели формы анкеты пользователя сайта, содержащей следующие поля: имя, город, специальность, пол, вопрос о частоте использования интернета, e-mail, согласие на отправку новостей сайта, короткое резюме. Комментарий к тексту класса формы из примера на рисунке 1.

Поля `name`, `city`, `job`, `gender` являются однострочными текстовыми полями ввода и создаются как экземпляры класса `forms.CharField`. Аргумент `label` задаёт тестовую подпись поля, `min_length` и `max_length` – ограничения на длину вводимого текста, используемые при валидации.

Поле `gender` задаёт две радиокнопки и создаётся как экземпляр класса `forms.ChoiceField`. Аргумент `widget` задаёт виджет `forms.Select`, который обеспечивает отображение этого поля как набора радиокнопок. Аргумент `choices` содержит список кортежей, задающих пары “значение”-“надпись” для радиокнопок одной группы. Эти параметры указываются как текстовые строки. Значения полей могут содержать не только числа, как в примере, но и другой текст. Аргумент `initial=1` задаёт выбранную по умолчанию кнопку, в данном случае - первую кнопку группы.

Поле `internet` задаёт раскрывающийся список и создаётся как экземпляр класса `forms.ChoiceField` без указания виджетов. Через аргумент `choices` задается кортеж кортежей, содержащий пары “значение”-“надпись” для отдельных строк списка. Аргумент `initial=1` задает выбор по умолчанию для первой строки списка.

Поле `notice` задаёт флажок и создаётся как экземпляр класса `forms.BooleanField`. В метод действия для этого поля передаётся логическое значение `True` или `False` в зависимости от того, выбран флажок или сброшен. Аргумент `required=False` необходим, когда у пользователя должна быть возможность не выбирать флажок. Поле `email` создается как экземпляр класса `forms.EmailField`. Это тип обеспечивает валидацию поля ввода текста в соответствии со стандартным форматом e-mail. Аргумент `min_length` задаёт дополнительное условие валидации – ограничение минимальной длины.

Поле `message` является экземпляром класса `forms.CharField`, как и другие поля ввода, но для него в качестве значения атрибута `widget` указан экземпляр класса `forms.Textarea`, что обеспечивает создание многострочного поля ввода. При создании экземпляра `forms.Textarea` передается аргумент `attrs`, содержащий словарь параметров, задающих число текстовых строк и столбцов.

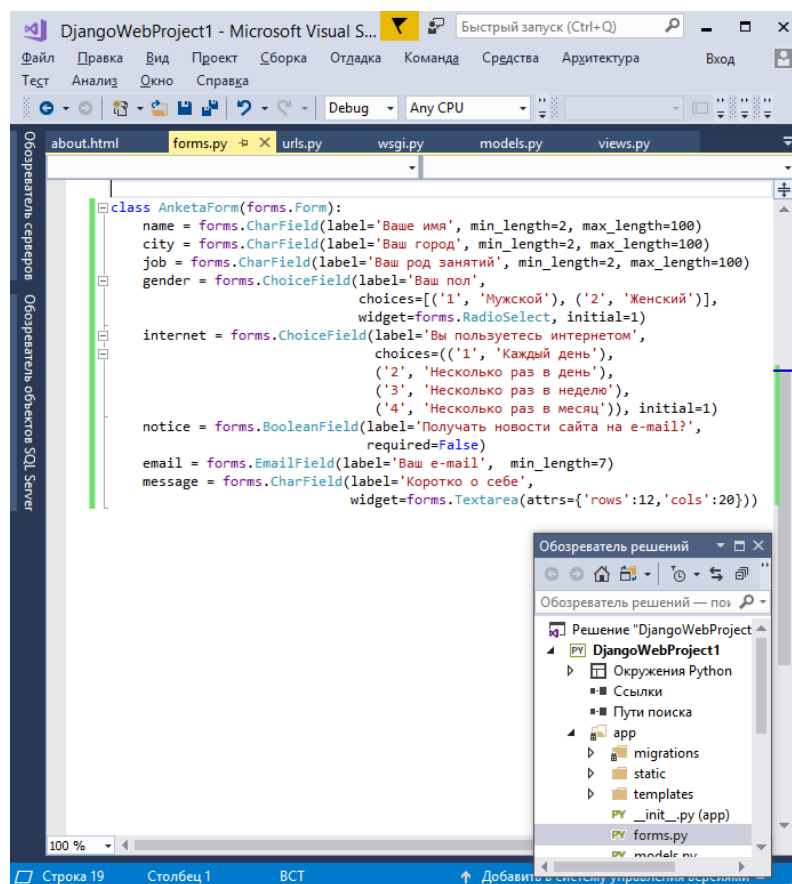


Рисунок 1 – Описание класса формы

1. Из теории в подразделе **Теория для практического занятия** следует, что после создания класса формы следует добавить разметку формы в файл шаблона HTML-страницы в папке **app\templates\app**. Добавление новой веб-страницы в проект Django рассматривалось в самостоятельной работе 4. Для этого требуется:
 - а) Добавить файл шаблона страницы в папку **app\templates\app**. В примере добавляется файл **anketa.html**. Разметка в файле построена таким образом, чтобы отображать форму в том случае, когда в шаблон передаётся переменная формы **form**, не равная **None** (пустому значению). В противном случае подразумевается, что переменная **data**, переданная в шаблон страницы, содержит поля для отображения результатов отправки формы, и выполняется отображение этих данных. Разметка шаблона страницы анкеты приведена на рисунке 2.

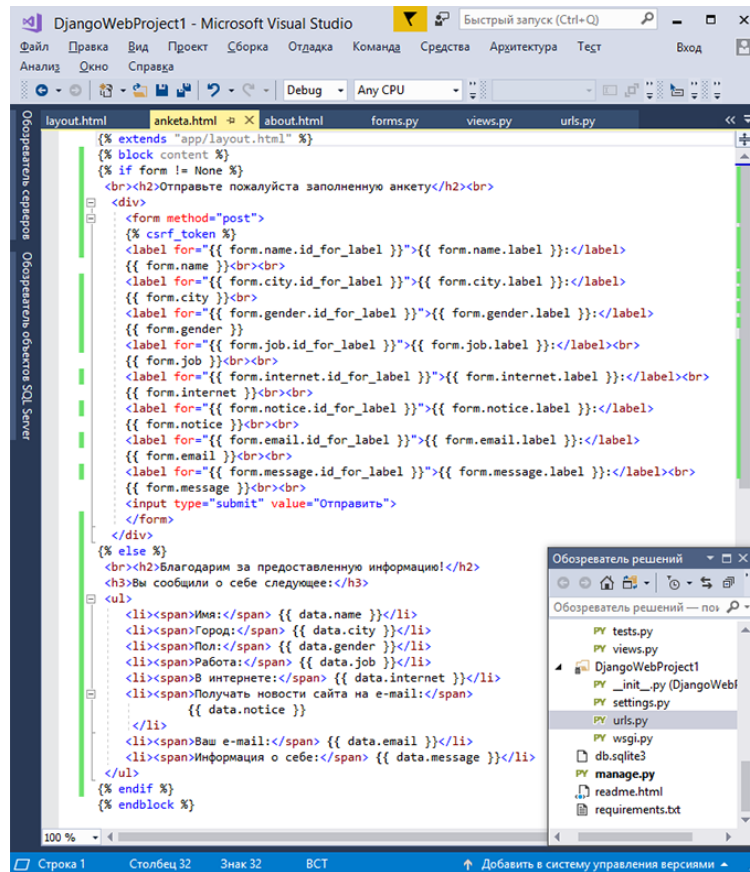


Рисунок 2 – Разметка шаблона страницы, содержащей форму

б) Добавить в файл **views.py** строку импорта класса формы. В данном примере это строка `from .forms import AnкетаForm`. Затем следует добавить в файл **views.py** метод, который будет обрабатывать запросы к добавленной странице. В примере добавляется метод `def anketa(request)`. Добавить в файл **views.py** метод, который будет обрабатывать запросы к добавленной странице. В примере добавляется метод `def anketa(request)`. Текст метода приведен на рисунке 3. В начале кода метода создаются словари **gender** и **internet**, используемые в дальнейшем для получения текста выбранных пользователем полей по их значению, передаваемому из формы. Далее, если в метод передана заполненная форма методом POST, выполняется валидация формы, и, в случае успешного результата, данные полей формы переносятся в словарь **data**. Если валидация пройдет неудачно, заполненная форма будет передана обратно в шаблон страницы для редактирования. Если же форма не была отправлена методом POST, то создаётся пустая новая форма для заполнения её пользователем. Затем вызывается метод **render**, которому в последнем аргументе передаётся словарь, содержащий переменные **form** и **data**.

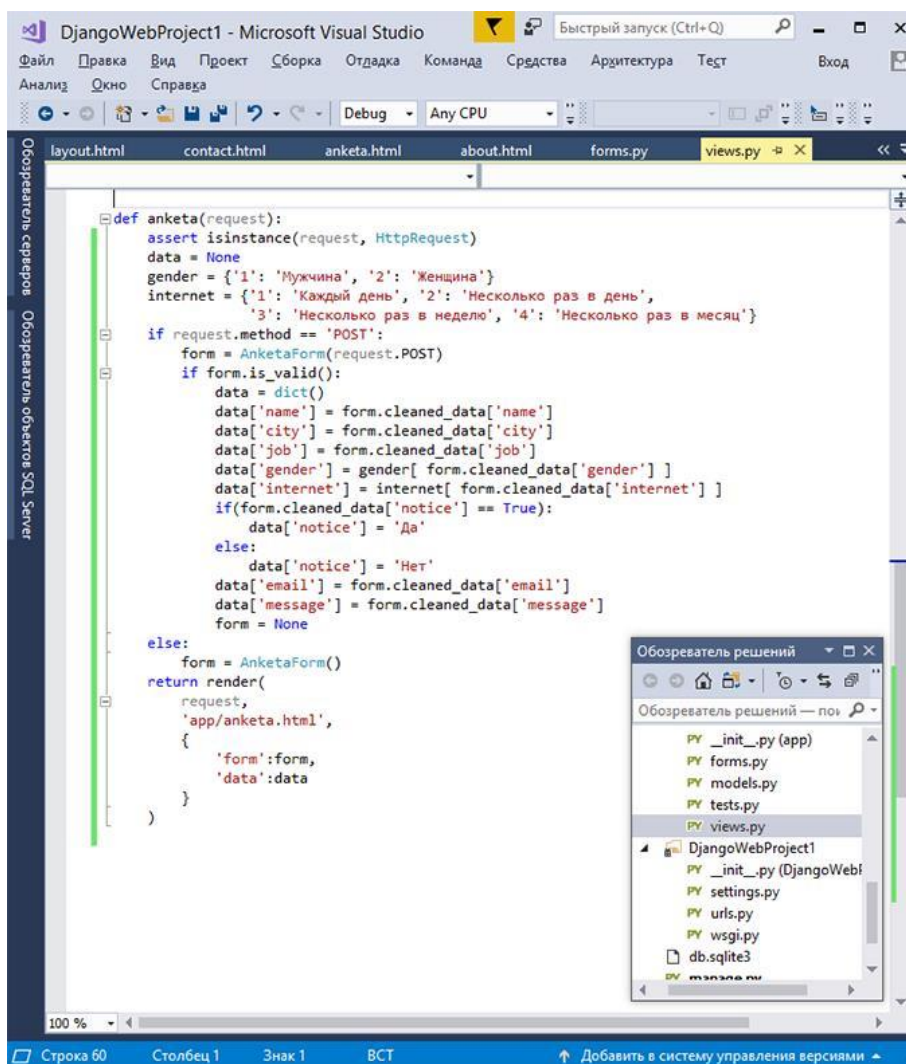


Рисунок 3 – Код метода, обрабатывающего запрос страницы с формой

в) В файл **templates\app\layout.html** добавить в список пунктов меню (тег ****) строку для добавляемой веб-страницы. В рассматриваемом примере это будет строка **Анкета**

г) В файл **urls.py** добавить URL для добавляемой веб-страницы. В рассматриваемом примере это будет строка

path('anketa/', views.anketa, name='anketa'),

На рисунке 4 представлен вид страницы с анкетой при переходе на неё пользователем через меню сайта. При оформлении страницы использованы стилевые правила из файла **site.css**.

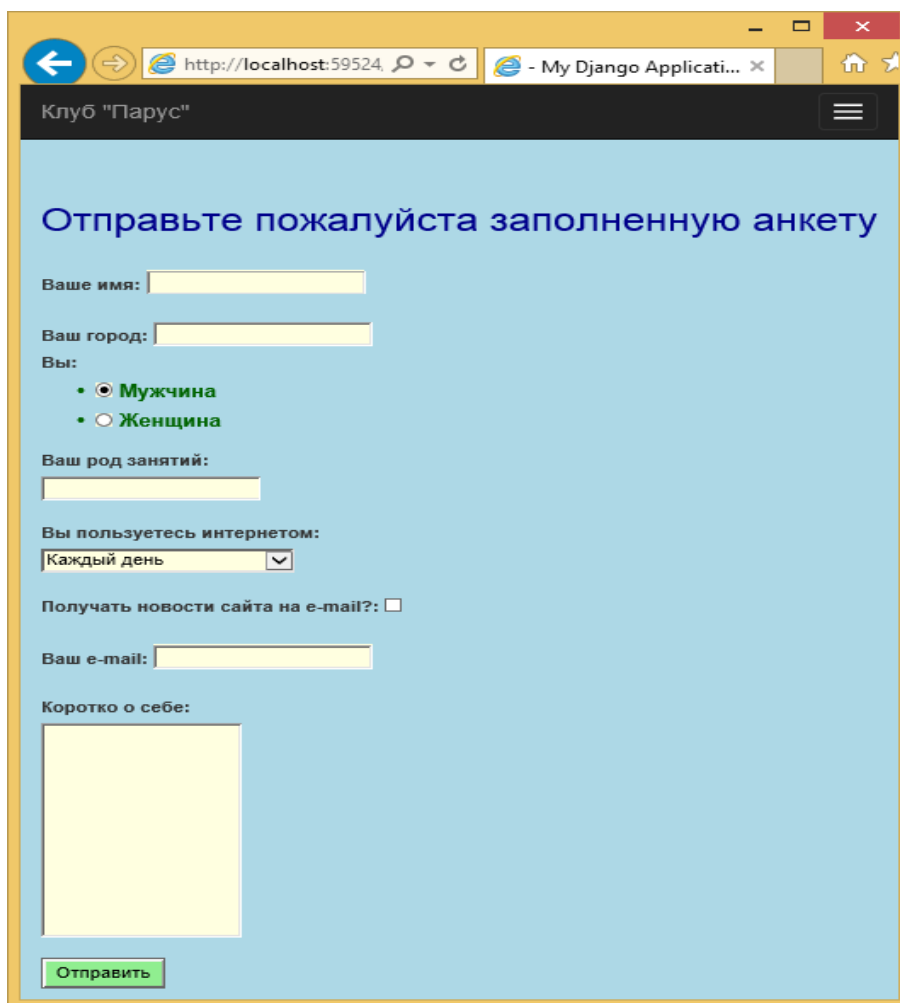


Рисунок 4 – Вид загруженной страницы с формой

На рисунке 5 приведен пример отображения информации на странице после успешной отправки анкеты на сервер.

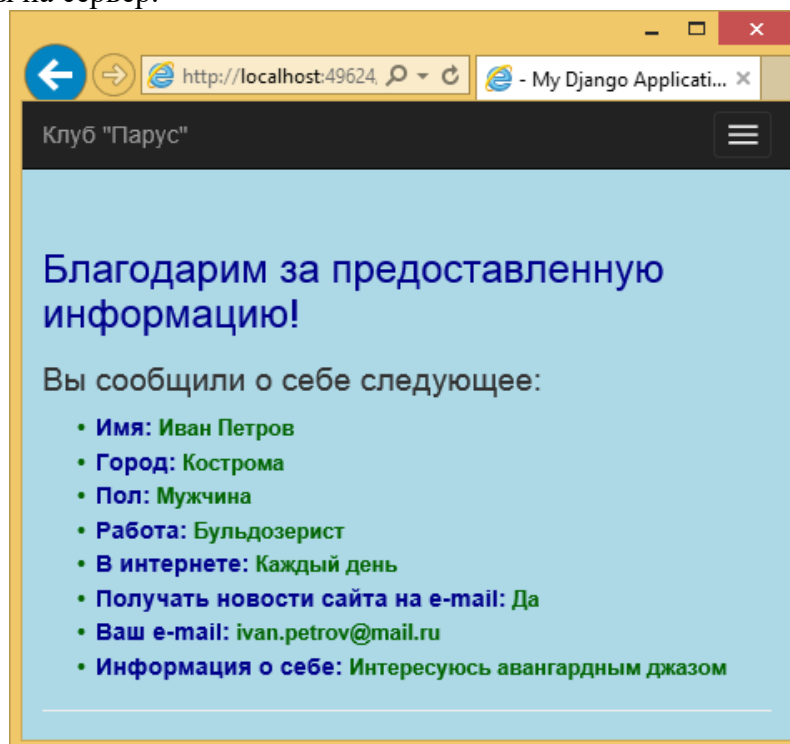


Рисунок 5 – Отображение информации об успешном приеме анкеты

Лабораторная работа 6. Создание администратора. Работа в административном разделе сайта. Регистрация пользователей

1. Открыть созданный веб-проект.
2. На основе материалов конспекта лекции 8 необходимо:
 - a. В среде Visual Studio в меню **Проект** — команда **Создание суперпользователя Django** (см. рисунок 1).

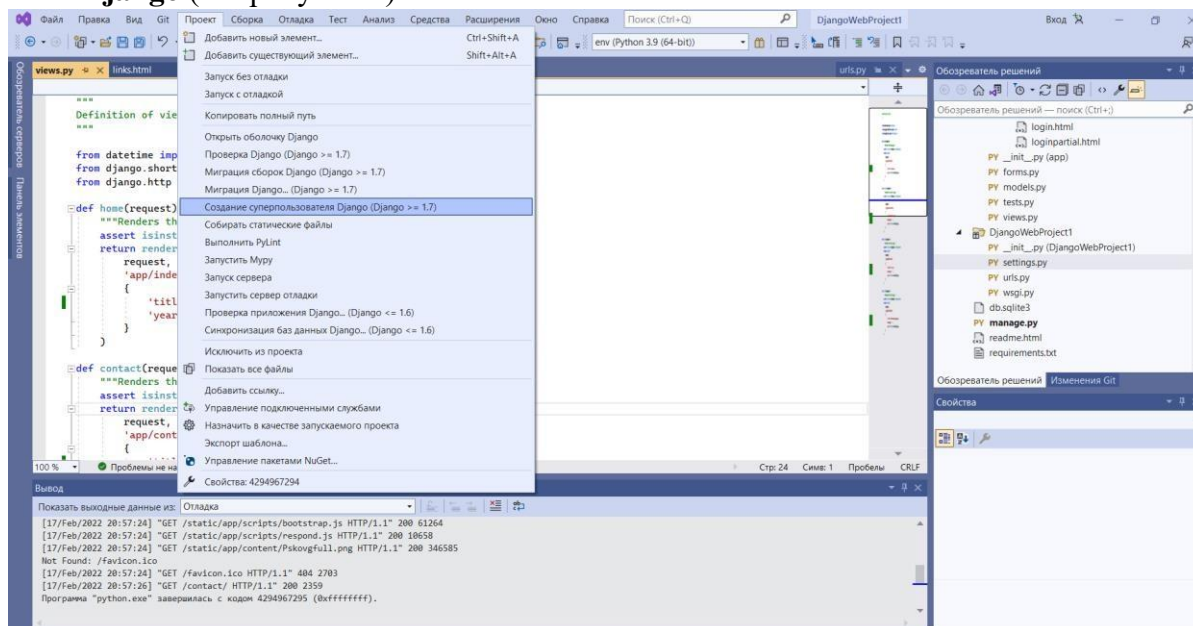


Рисунок 1 – Выбор команды «Создание суперпользователя Django»

- b. В консоли Django требуется ввести имя суперпользователя – например, *admin*, его электронную почту, пароль пользователя *admin* и подтверждение пароля (например, см. рисунок 2).

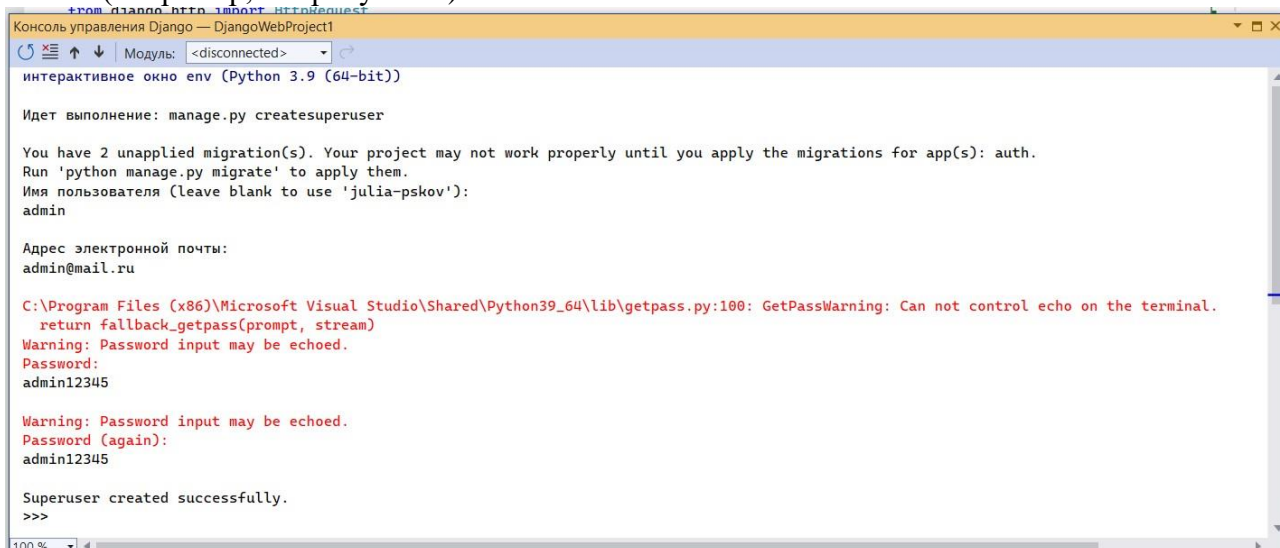


Рисунок 2 – Ввод данных суперпользователя Django

4. На основе материалов конспекта лекции 8 необходимо:
 - a. Запустить проект на выполнение;
 - b. Добавить в строке браузера **admin**;
 - c. В форме входа в административный раздел ввести имя пользователя (логин) - **admin** и его пароль (см. рисунок 3) и войти в административный раздел;

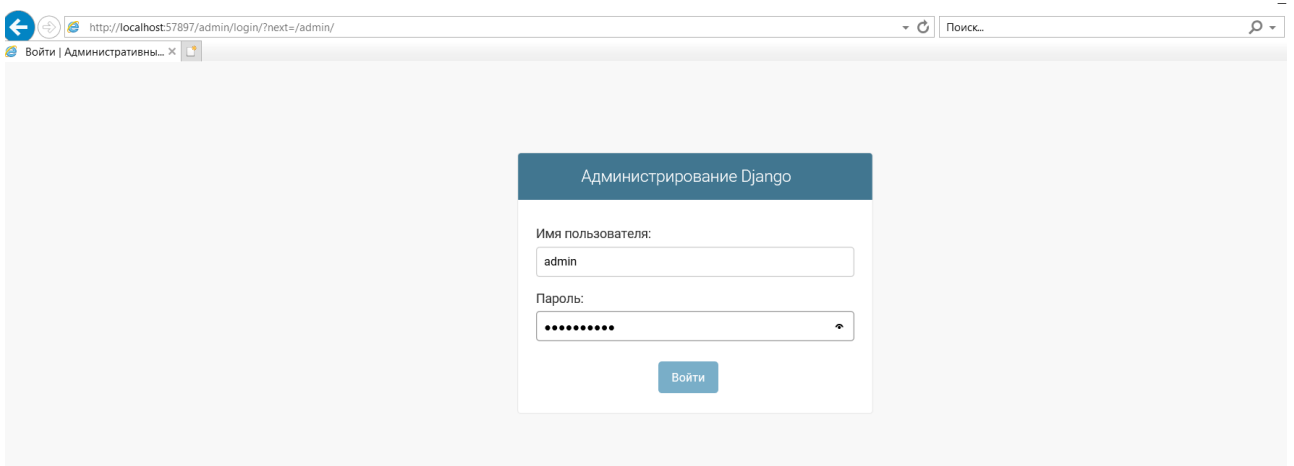


Рисунок 3 – Ввод данных администратора в форму входа в административный раздел
 д. Выбрать группу **Пользователи**;
 е. Посмотреть данные администратора – выбрать **admin** (см. рисунок 4);

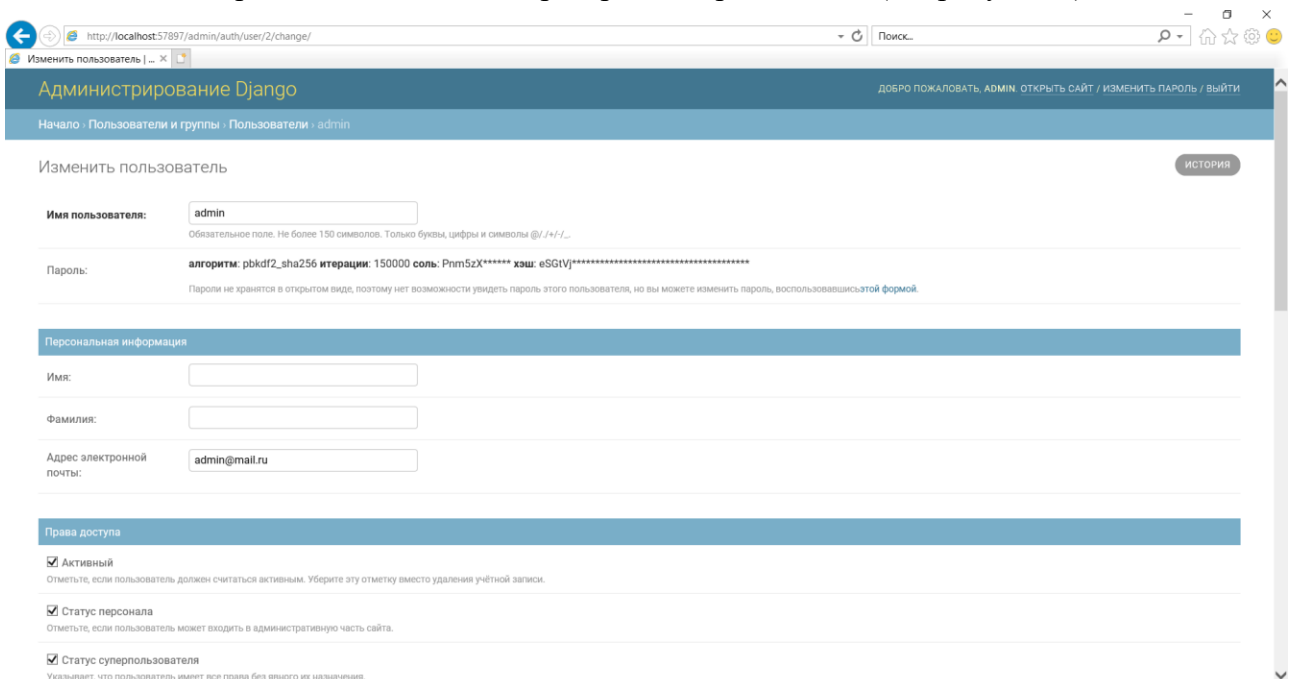


Рисунок 4 – Просмотр данных администратора в административном разделе
 ф. Выйти из административного раздела.

II Рассмотрим пример решения задания 2 практического занятия по теме 6.

1. Из теории в подразделе **Теория для практического занятия** следует, что для того, чтобы добавить веб-страницу регистрации пользователей, необходимо:
 - а. В файле **views.py** необходимо добавить строку:
`from django.contrib.auth.forms import UserCreationForm`
 - б. В файле **views.py** добавьте метод действия контроллера **registration**:

```

def registration(request):
    """Renders the registration page."""

    if request.method == "POST":
        regform = UserCreationForm(request.POST)
        if regform.is_valid():
            reg_f = regform.save(commit=False)
            reg_f.is_staff = False
            # запрещен вход в
            # административный раздел
            reg_f.is_active = True
            # активный пользователь
            reg_f.is_superuser = False
            # не является суперпользователем
            reg_f.date_joined = datetime.now()
            # дата регистрации
            reg_f.last_login = datetime.now()
            # дата последней авторизации

            regform.save()
            # сохраняем изменения после добавления полей

            return redirect('home')
            # переадресация на главную страницу после
            # регистрации
        else:
            regform = UserCreationForm()
            # создание объекта формы для ввода данных

    assert isinstance(request, HttpRequest)
    return render(
        request,
        'app/registration.html',
        {
            'regform': regform,
            # передача формы в шаблон веб-страницы

            'year': datetime.now().year,
        }
    )

```

- c. В файле **views.py** дополните строку импорта функции *render* еще и импортом функции *redirect*:
- ```
from django.shortcuts import render, redirect
```
- d. Добавьте в папку проекта **app\templates\app\** файл шаблона веб-страницы **registration.html** для отображения веб-страницы с формой регистрации пользователя (см. рисунок 5, рисунок 6).

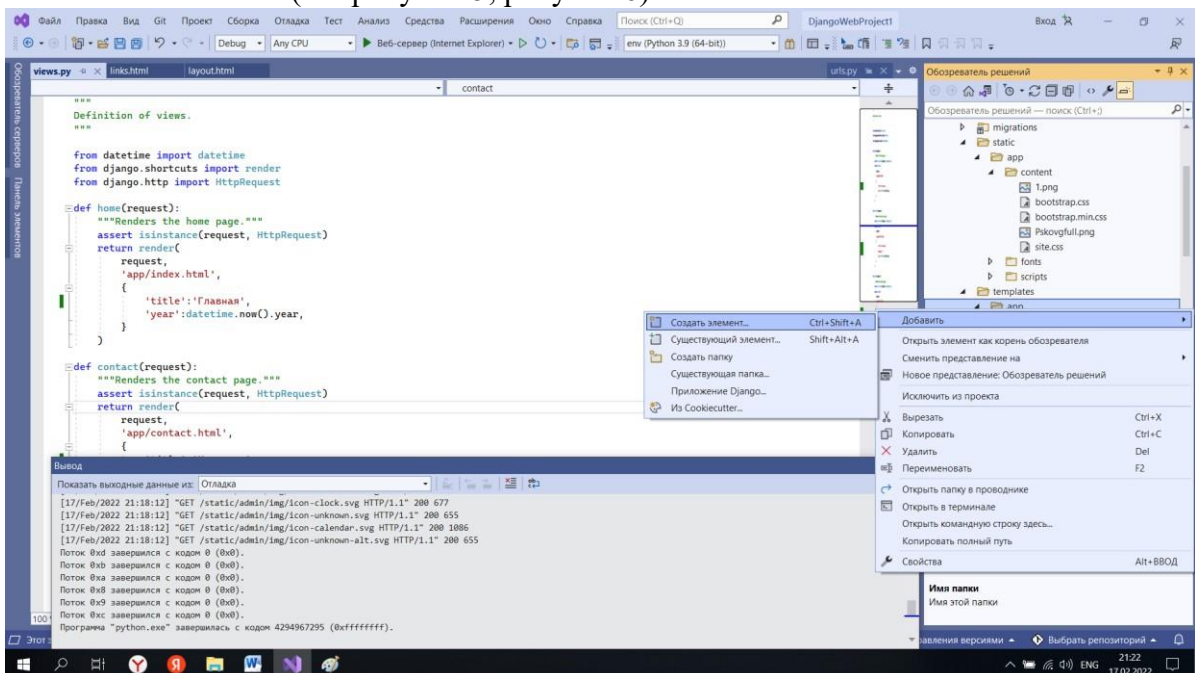


Рисунок 5 – Добавить новый элемент в веб-проект

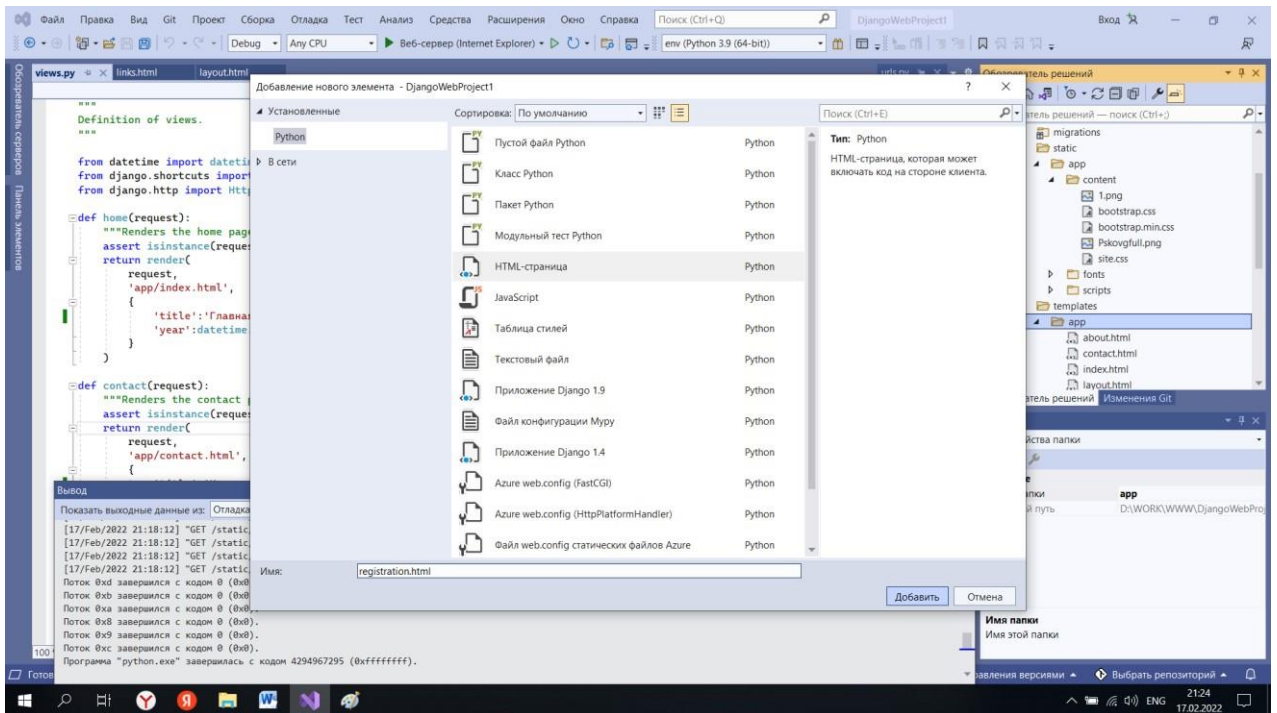


Рисунок 6 – Добавить registration.html в веб-проект

е. В файле **registration.html** необходимо заменить содержимое на код отображения формы регистрации:

```
{% extends "app/layout.html" %}
{% block content %}
<h2>Регистрация пользователя</h2>
 <form method="POST" > {% csrf_token %}
 <div>
 {{ regform.as_p}}

 <button type="submit" class="save btn btn-default" >
Зарегистрировать </button>
 </div>
 </form>
{% endblock %}
```

ф. В файл **urls.py** необходимо добавить **URL** для веб-страницы **registration**:  
 path('registration/', views.registration, name='registration'),

г. В файл **layout.html** в блок с пунктами меню добавить новый пункт **Регистрация**, который должен быть доступен для не авторизованных пользователей:

```
{% if not user.is_authenticated %}
Регистрация
{% endif %}
```

2. Из теории в подразделе **Теория для практического занятия** следует, что для того, чтобы зарегистрировать тестового пользователя, необходимо:

- a. Запустить проект на выполнение.
- b. На странице **Регистрация** ввести данные тестового пользователя в форму и нажать кнопку **Зарегистрировать** (см. рисунок 7).

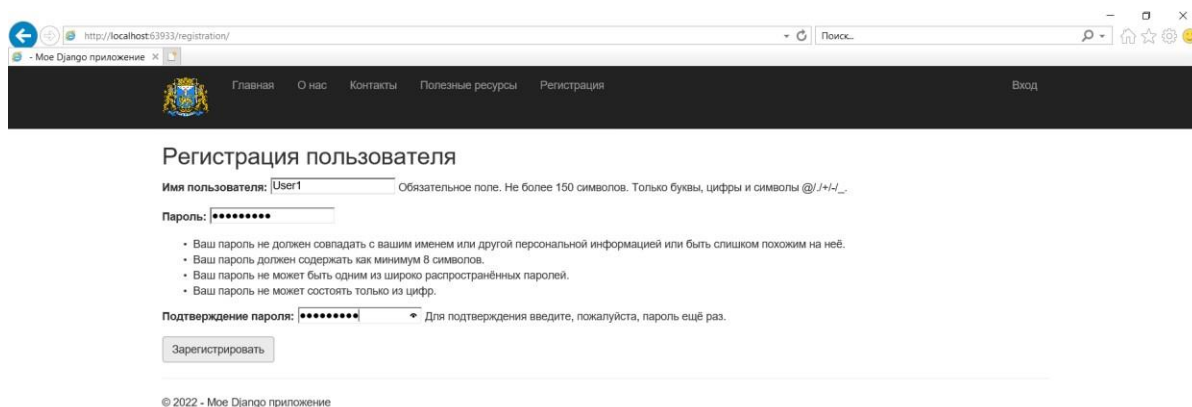


Рисунок 7 – Регистрация тестового пользователя

3. Из теории в подразделе **Теория для практического занятия** следует, что для того, чтобы изменить англоязычные названия на веб-странице **Log in (Вход)** на русскоязычные, необходимо:

- а. В файлах **loginpartial.html** и **login.html** заменить англоязычные названия приветствия, ссылок, меток, кнопки, сообщений на русскоязычные (см. рисунок 8, рисунок 9):

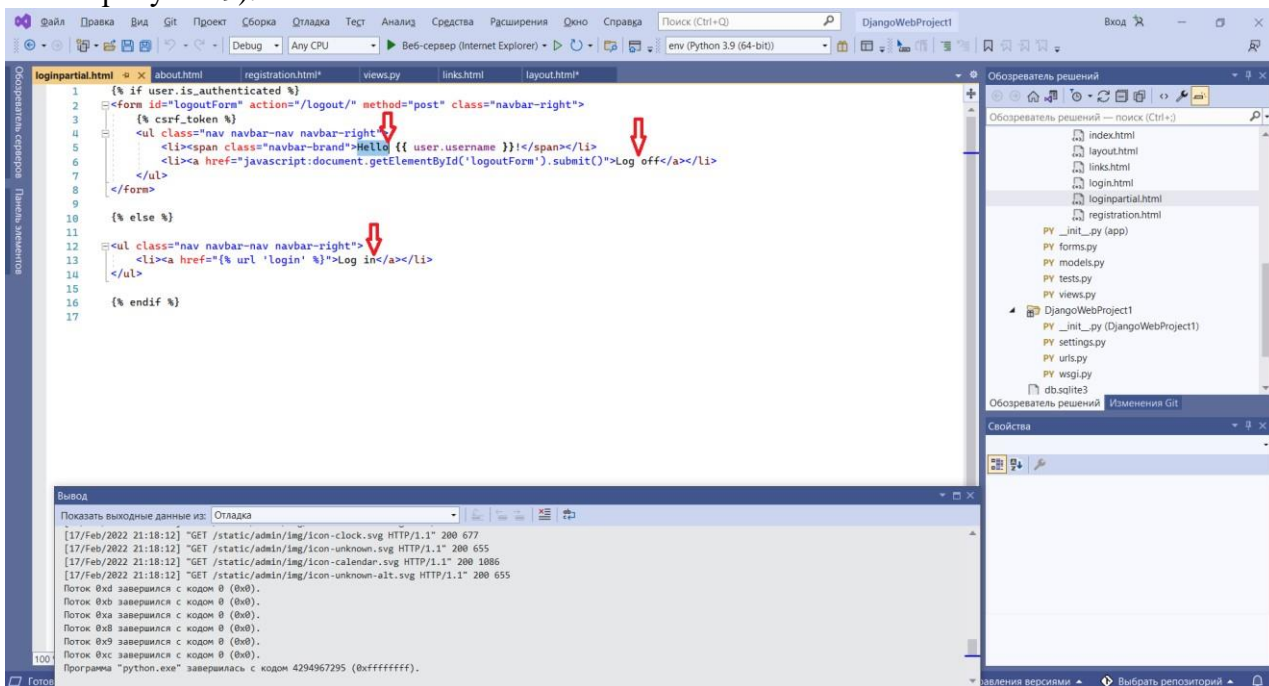


Рисунок 8 – Замены в файле loginpartial.html

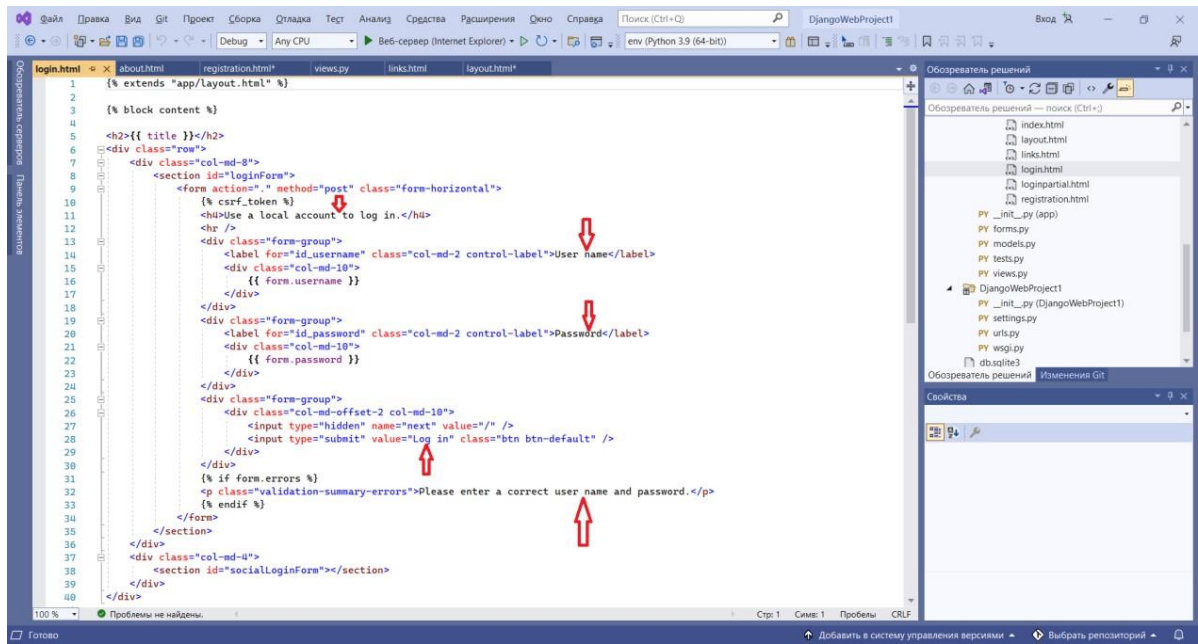


Рисунок 9 – Замены в файле login.html

б. В файле urls.py заменить заголовок **Log in**, например, на **Авторизация** (см. рисунок 10):

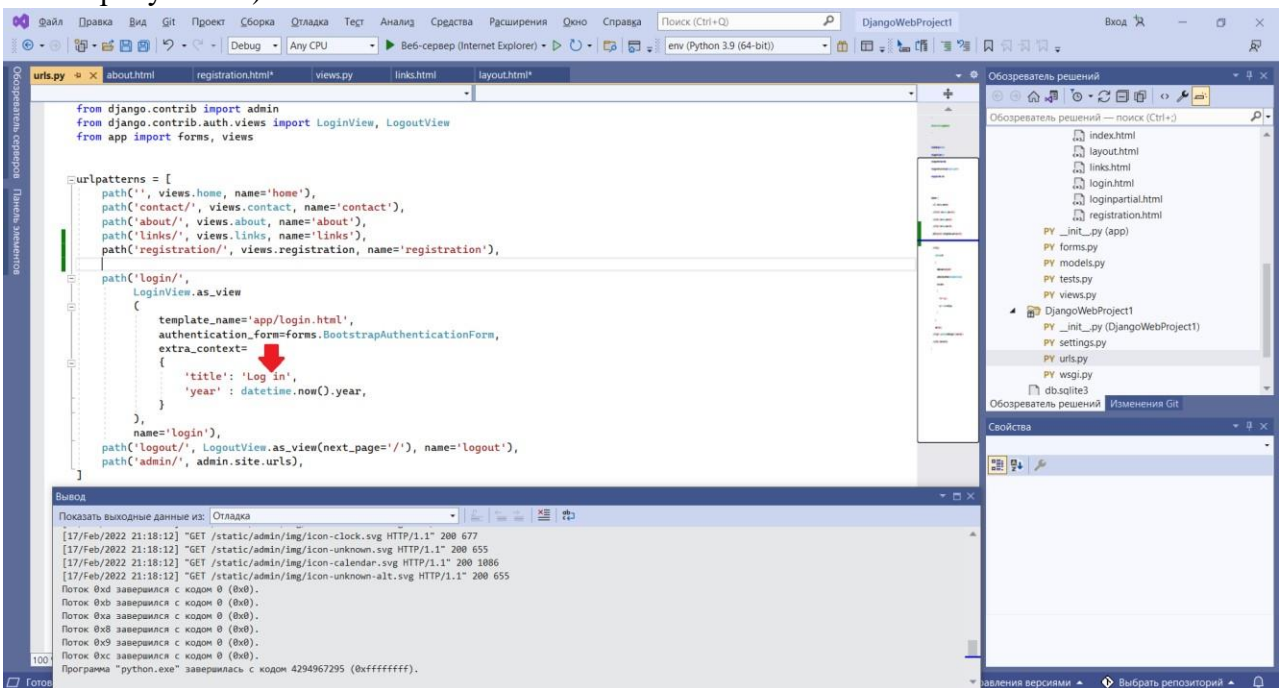


Рисунок 10 – Замена в файле urls.py

с. В файле forms.py англоязычные надписи полей формы заменить на русскоязычные (см. рисунок 11):

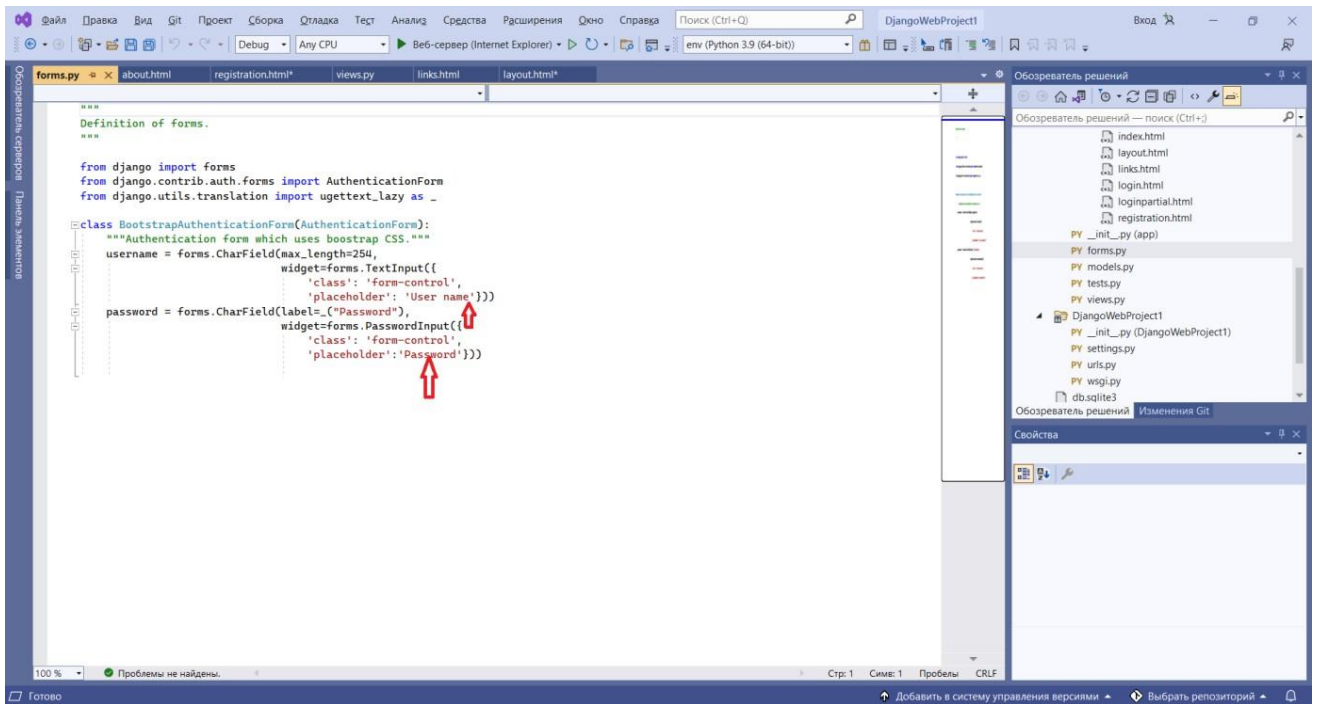


Рисунок 11 – Замены в файле forms.py

4. Для того, чтобы проверить вход тестового пользователя на сайт, необходимо:

- a. Запустить проект на выполнение.
- b. Выбрать пункт меню **Вход**.
- c. На странице **Вход** ввести **Логин** и **Пароль** тестового пользователя и нажать кнопку **Войти** (см. рисунок 12).

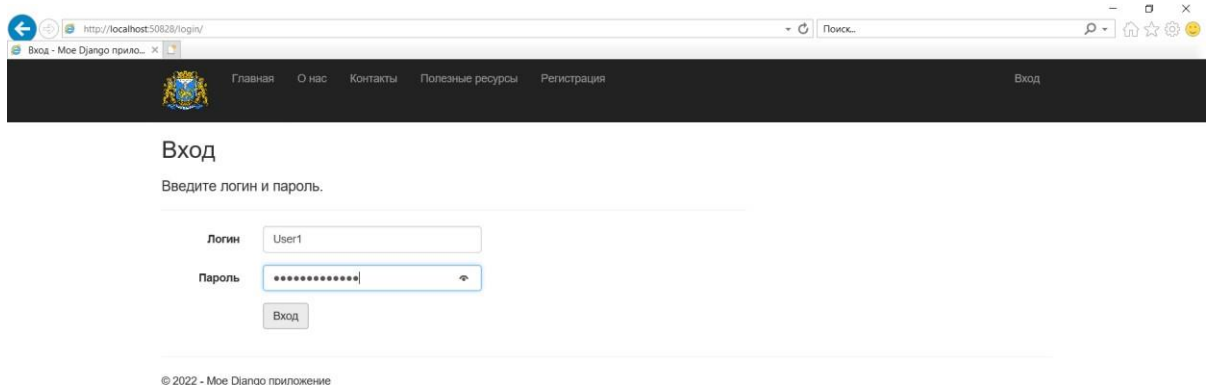


Рисунок 12 – Вход тестового пользователя на сайт

## Лабораторная работа 7. Создание модели данных. Организация взаимодействия с базой данных

необходимо:

- a. Открыть, созданный ранее, веб-проект Django в Visual Studio. В нём открыть файл **models.py**.
- b. В файл **models.py** добавить код класса модели блога – **Blog**, содержащий:

- Поля модели для *Заголовка*, *Краткого содержания*, *Полного содержания* и *Даты публикации статьи блога* (см. рисунок 1).
  - Методы `get_absolute_url` и `__str__` (см. рисунок 1).
  - Метаданные – `class Meta` (см. рисунок 1).
- с. В начало файла `models.py` необходимо добавить строку импорта класса `datetime` из модуля `datetime` (см. рисунок 1).
- д. В начало файла `models.py` добавить строку импорта метода `reverse` (см. рисунок 1).
- е. В начало файла `models.py` необходимо добавить строку импорта модуля административного раздела (см. рисунок 1).
- ф. В конец файла `models.py` после объявления модели (`class Blog`) необходимо добавить строку регистрации модели блога в административном разделе (см. рисунок 1):
- ```
admin.site.register(Blog)
```

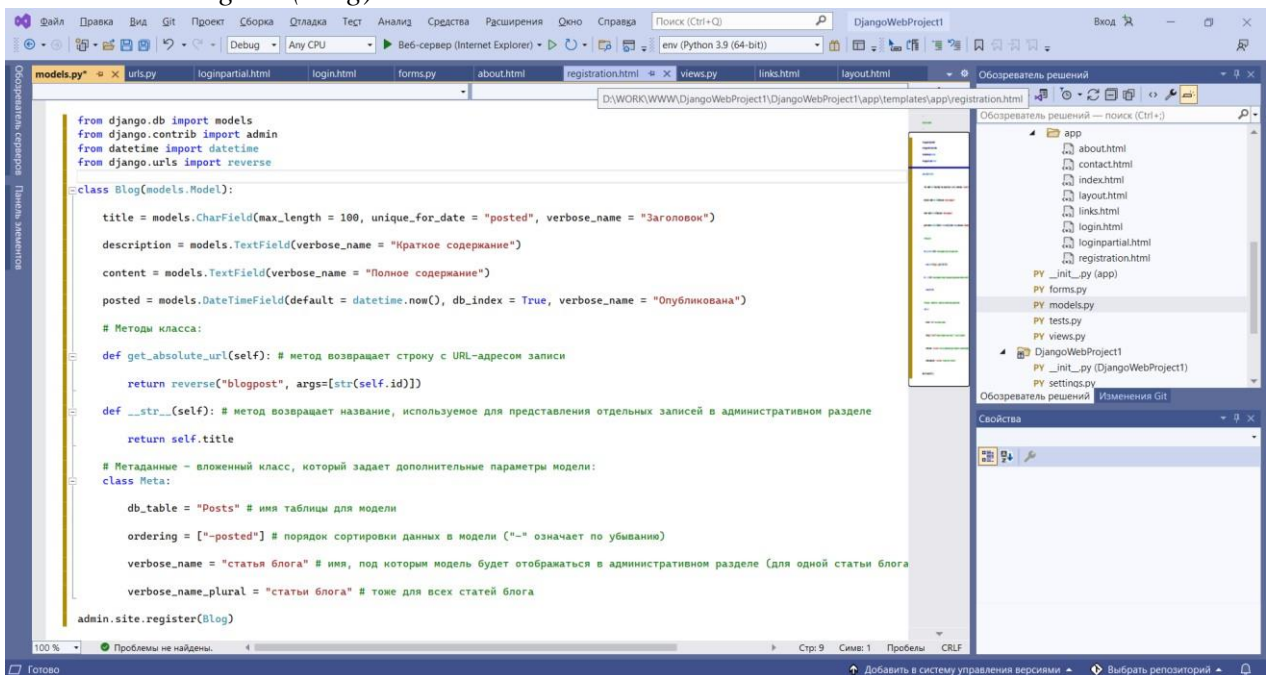


Рисунок 1 – Добавление в веб-проект модели статьи блога

2. На основе материалов лекции 9 необходимо:

а. В меню **Проект** выбрать команду **Миграция сборки Django**

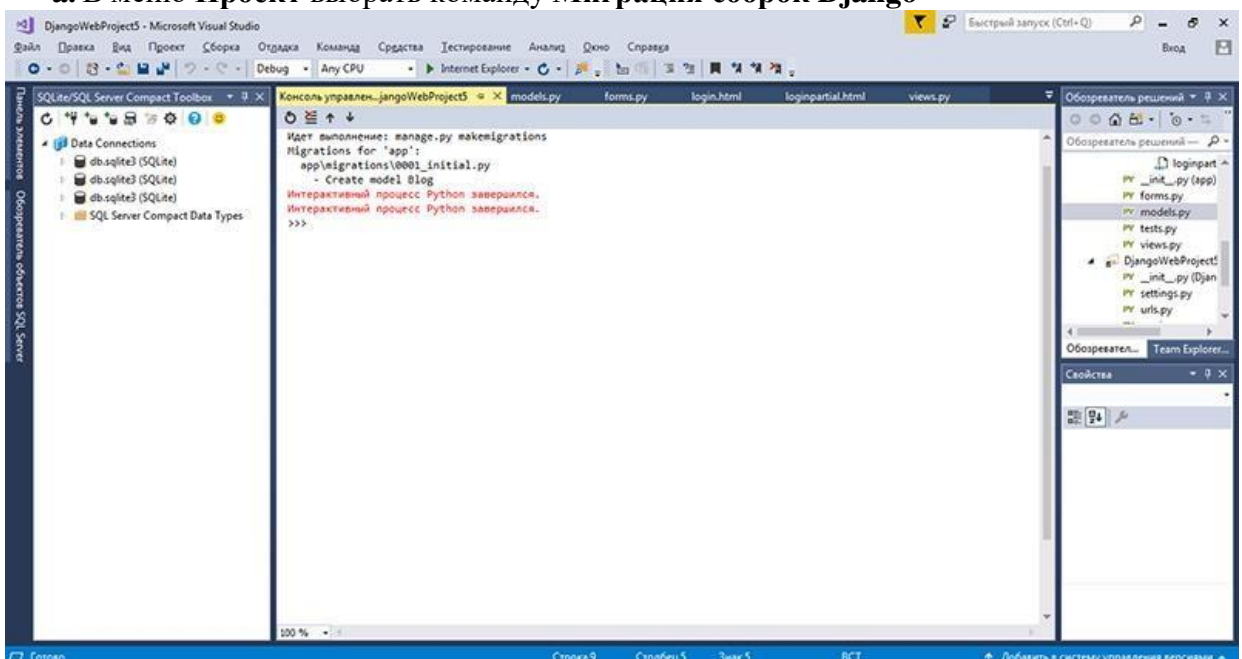


Рисунок 2 – Результат выполнения команды Миграция сборок Django

в. В меню Проект выбрать команду Миграция Django...

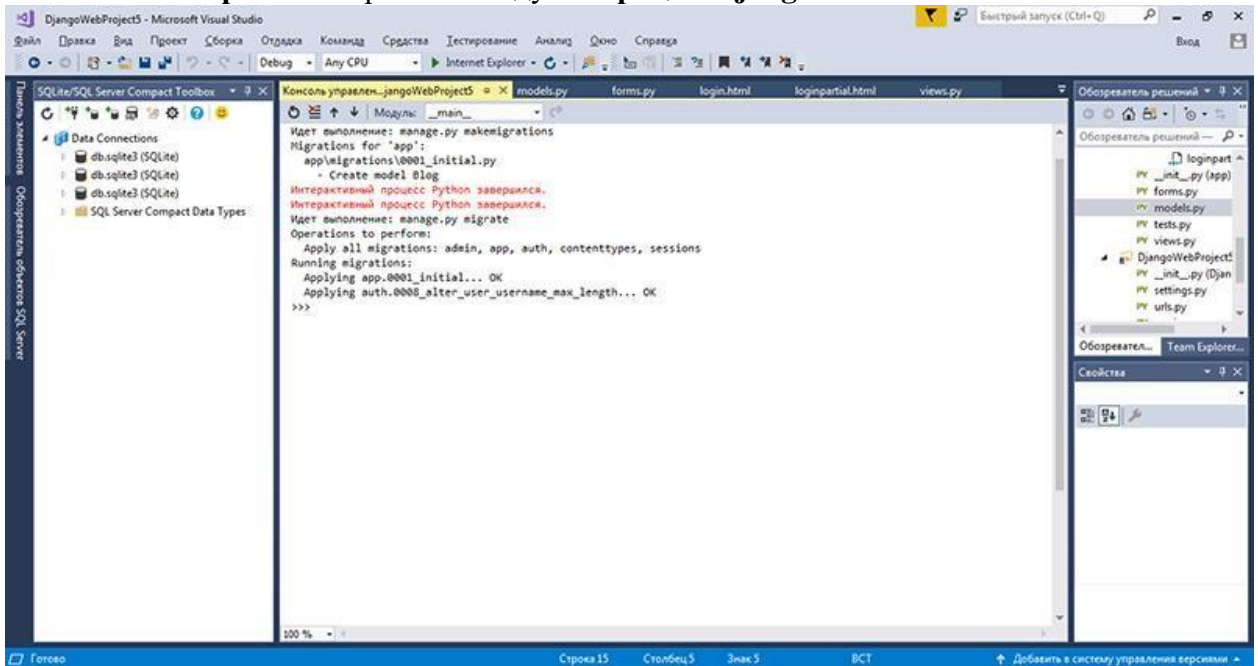


Рисунок 3 – Результат выполнения команды Миграция Django...

3. На основе теории в подразделе **Теория для практического занятия** необходимо:

- а. Запустить проект на выполнение.
- б. Войти в административный раздел (под учетной записью ранее созданного администратора).
- с. В строке «**Статьи блога**» нажать кнопку «+ **Добавить**» (см. рисунок 4).

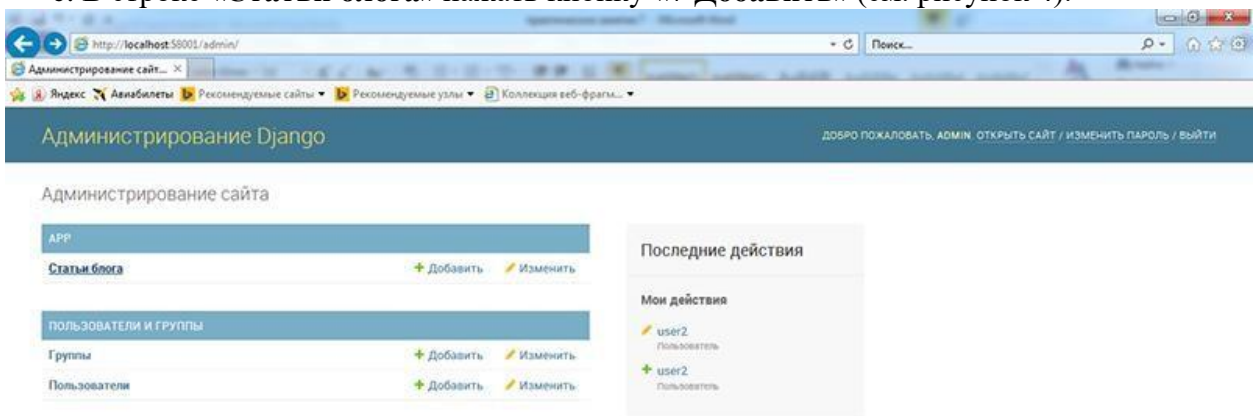


Рисунок 4 – Добавление статьи блога

- д. В форме добавления статьи блога необходимо ввести *Заголовок статьи блога*, *Краткое содержание статьи*, *Полное содержание статьи*, дату публикации статьи – поле *Опубликована* (см. рисунок 5)

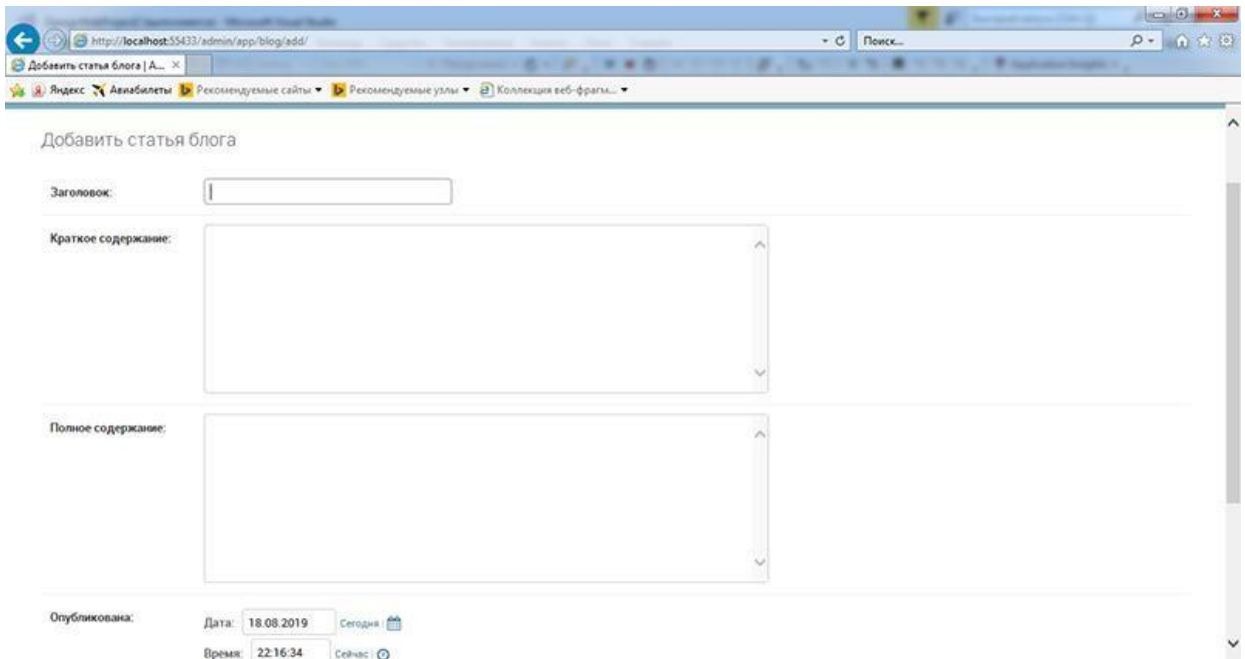


Рисунок 5 – Заполнение формы Добавления статьи блога е. Нажать кнопку «Сохранить» (результат на рисунке б).

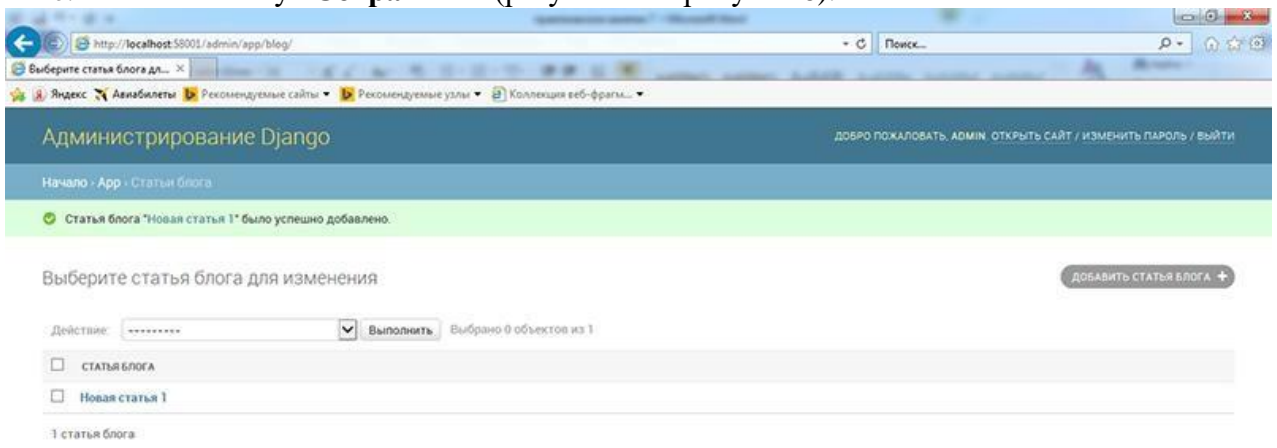
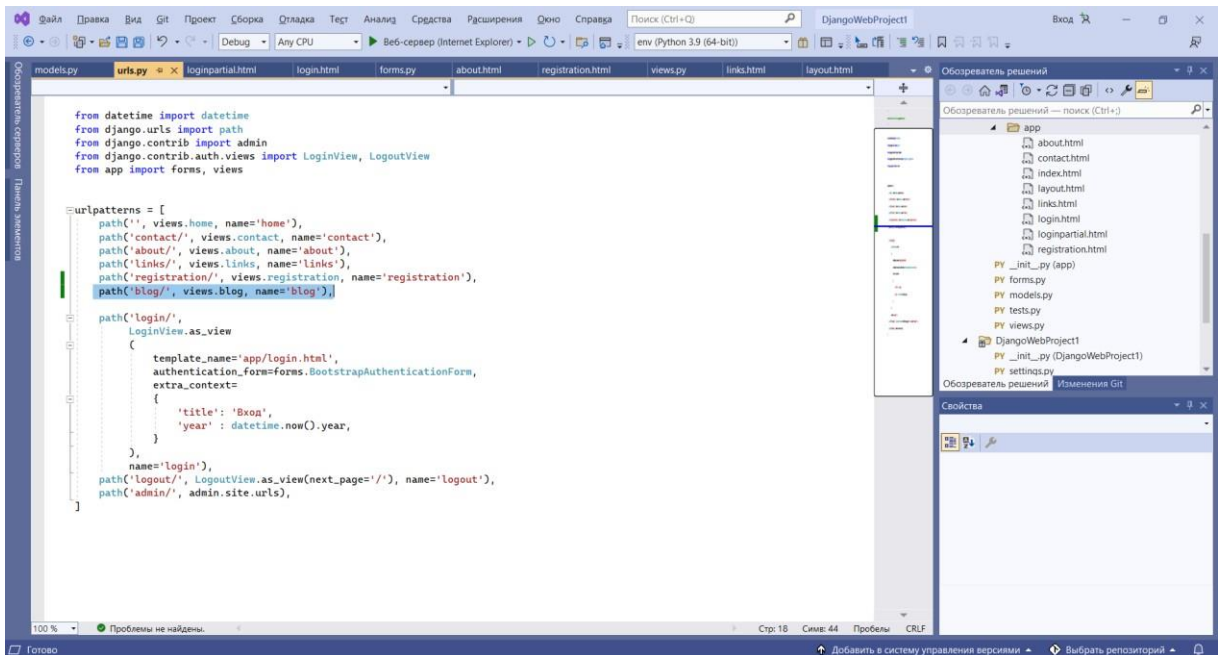


Рисунок 6 – Добавленная статья в списке статей блога

- f. Добавить следующую новую статью блога, нажав на кнопку "**ДОБАВИТЬ СТАТЬЯ БЛОГА +**"
- g. В форме ввести данные новой статьи и сохранить в таблице базы данных (пункты d, e).
- h. Добавить следующую новую статью блога, нажав на кнопку "**ДОБАВИТЬ СТАТЬЯ БЛОГА +**"
- i. В форме ввести данные новой статьи и сохранить в таблице базы данных (пункты d, e).
- j. Выйти из административного раздела, нажав на ссылку **ВЫЙТИ**.

Лабораторная работа 8. Создание блога

1. необходимо:
 - a. Открыть, созданный ранее, веб-проект Django в Visual Studio.
 - b. В файл urls.py добавить шаблон URL для веб-страницы со списком постов – см. рисунок 1.



- Рисунок 1 – Добавление в веб-проект шаблона URL для страницы со списком постов с. В файл views.py добавить строки для использования модели Blog – см. рисунок 2.
 d. В файл views.py добавить метод действия контроллера blog – см. рисунок 3.
 e. В папку app\templates\app\ добавить файл blog.html для отображения списка статей блога – см. рисунок 4.
 f. В файле templates\app\layout.html добавить в список пунктов меню пункт со ссылкой на блог – см. рисунок 5.
 g. Проверить отображение статей блога на сайте – см. рисунок 6.

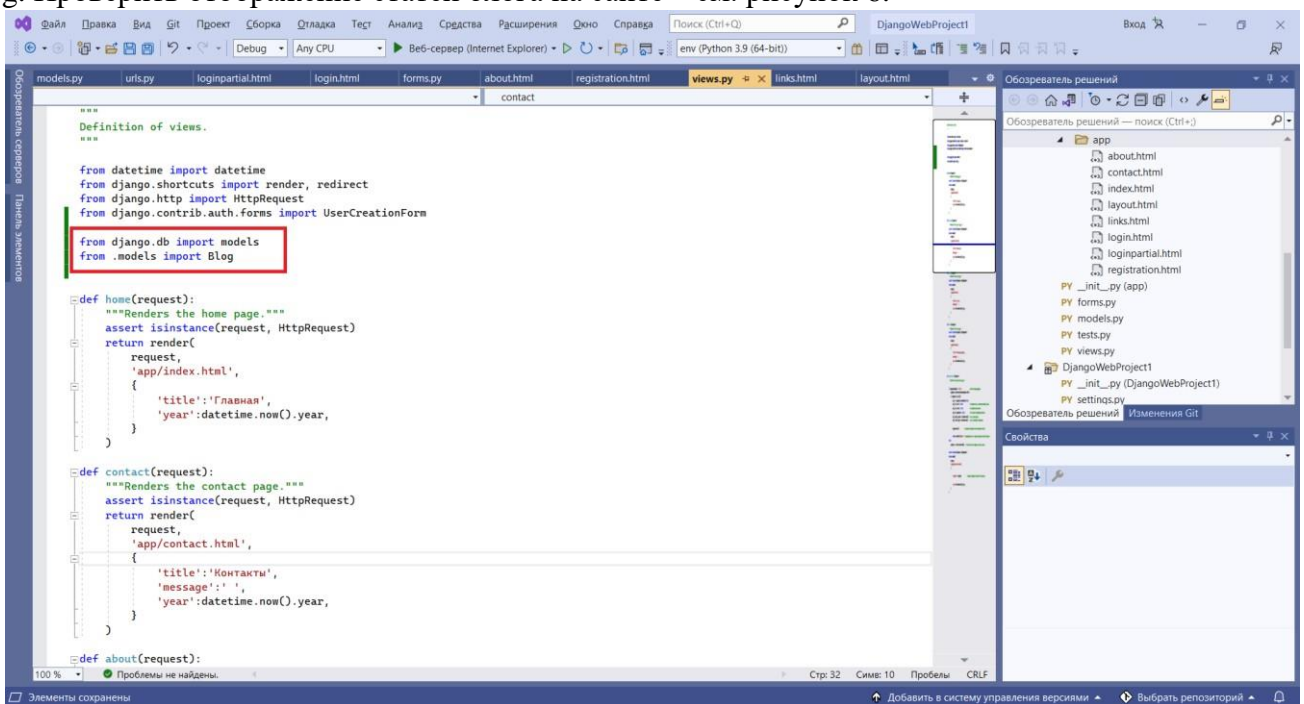


Рисунок 2 –Добавление строки использования модели Blog

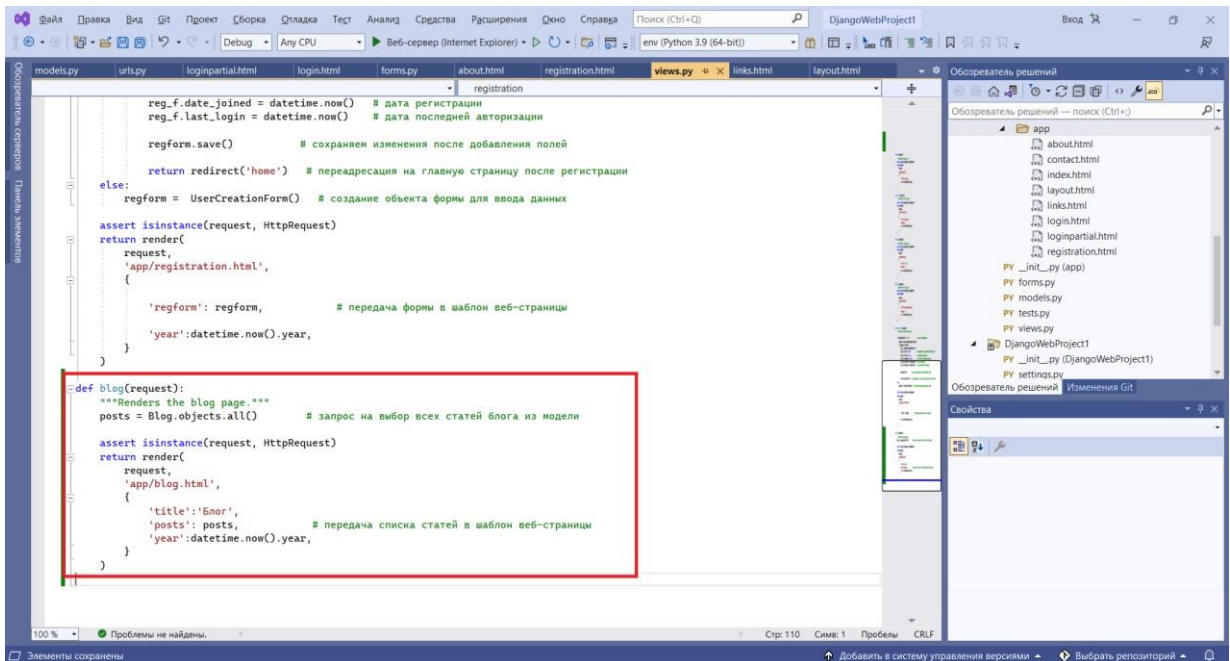


Рисунок 3 –Код метода контроллера blog

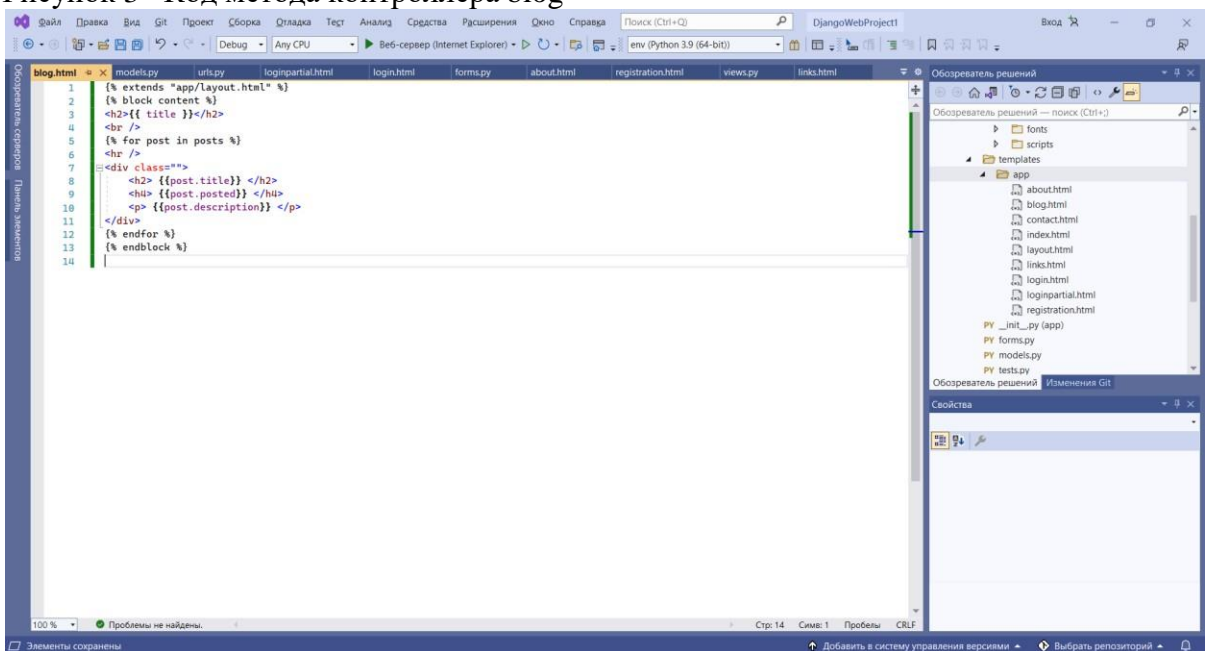


Рисунок 4 –Код веб-страницы blog.html

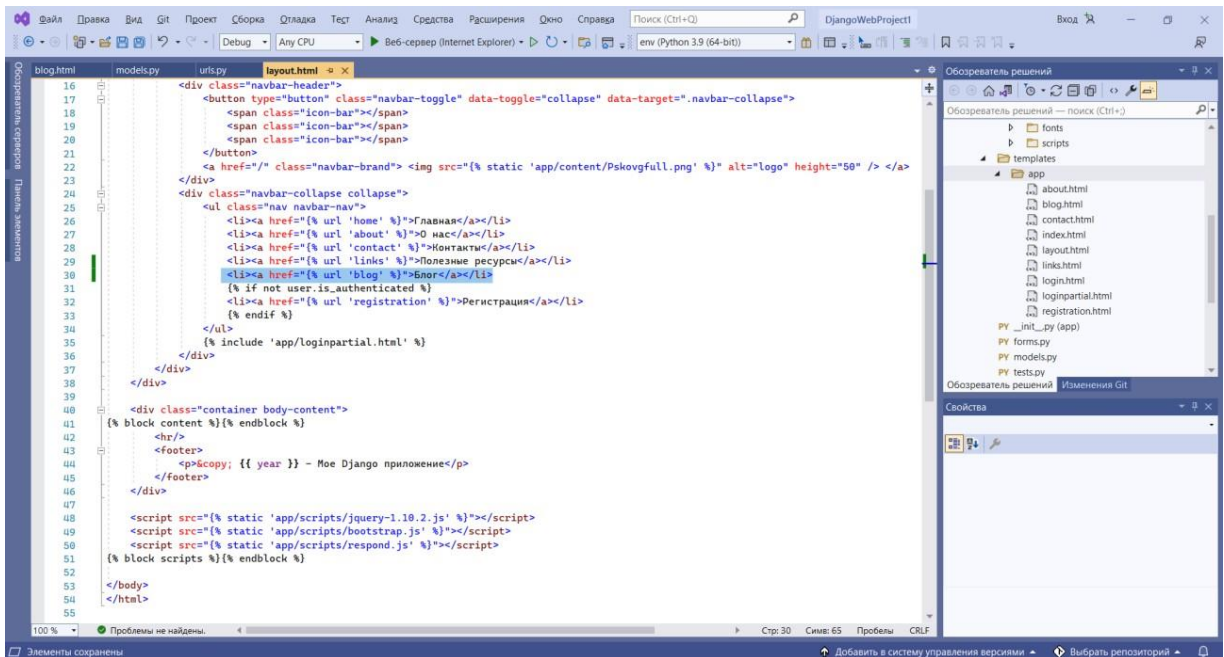


Рисунок 5 –Пункт в меню для веб-страницы blog.html

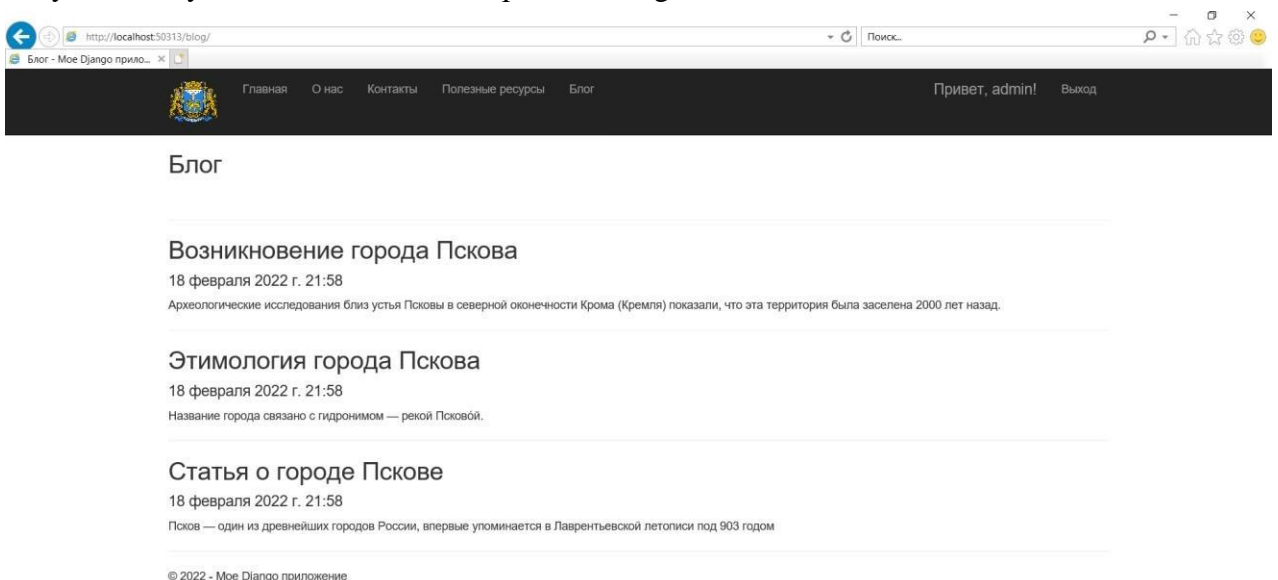


Рисунок 6 –Отображение статей блога на сайте

2. На основе теории в подразделе **Теория для практического занятия** необходимо:
 - а. В файл urls.py в шаблонах URL добавить строку для перехода на страницу поста по параметру – см. рисунок 7.
 - б. В файл blog.html добавить строку (после строки вывода краткого содержания статьи) для перехода на страницу поста по ссылке «Читать далее» – см. рисунок 8.

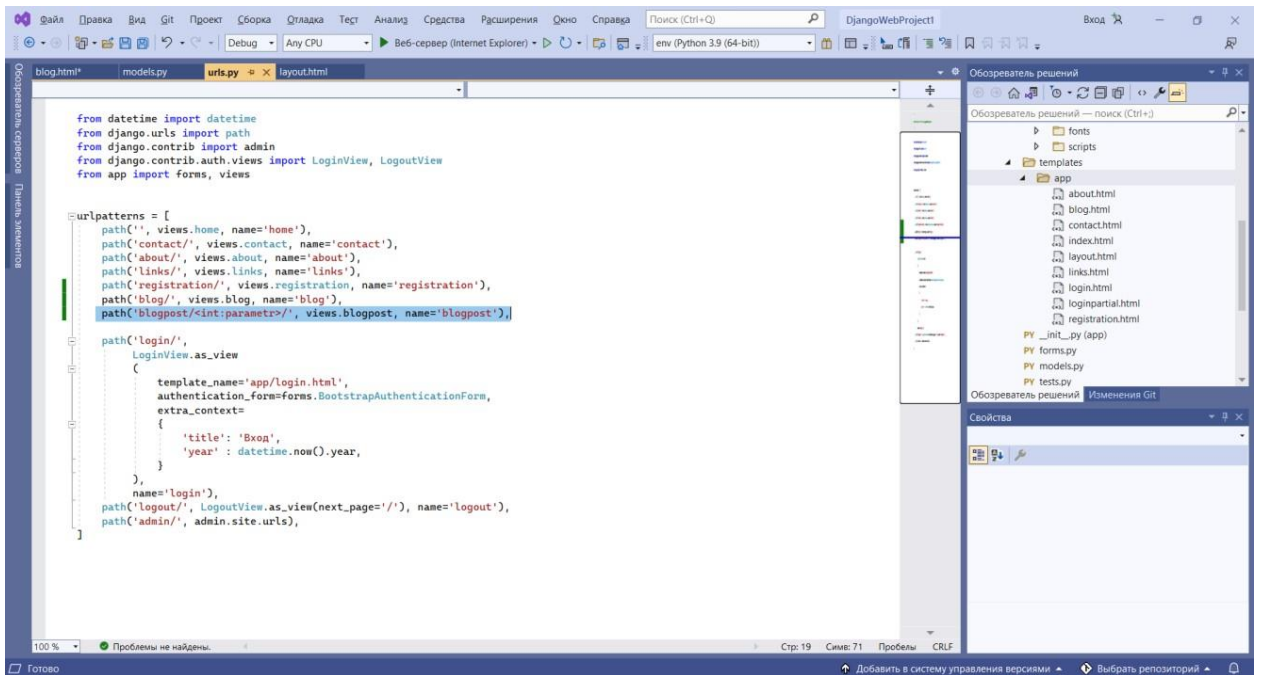


Рисунок 7 –Добавление URL для страницы поста

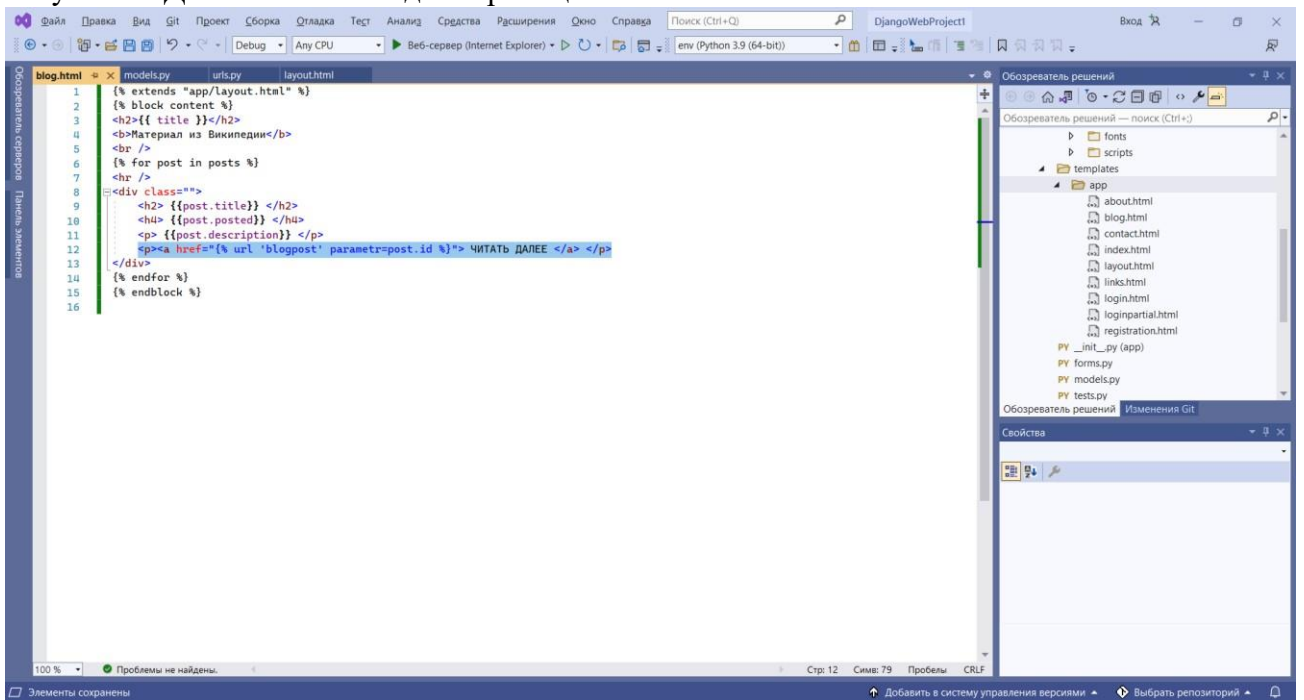


Рисунок 8 –Добавление ссылки на страницу поста

с. В файл `views.py` добавили код метода действия контроллера для страницы поста (см. рисунок 9).

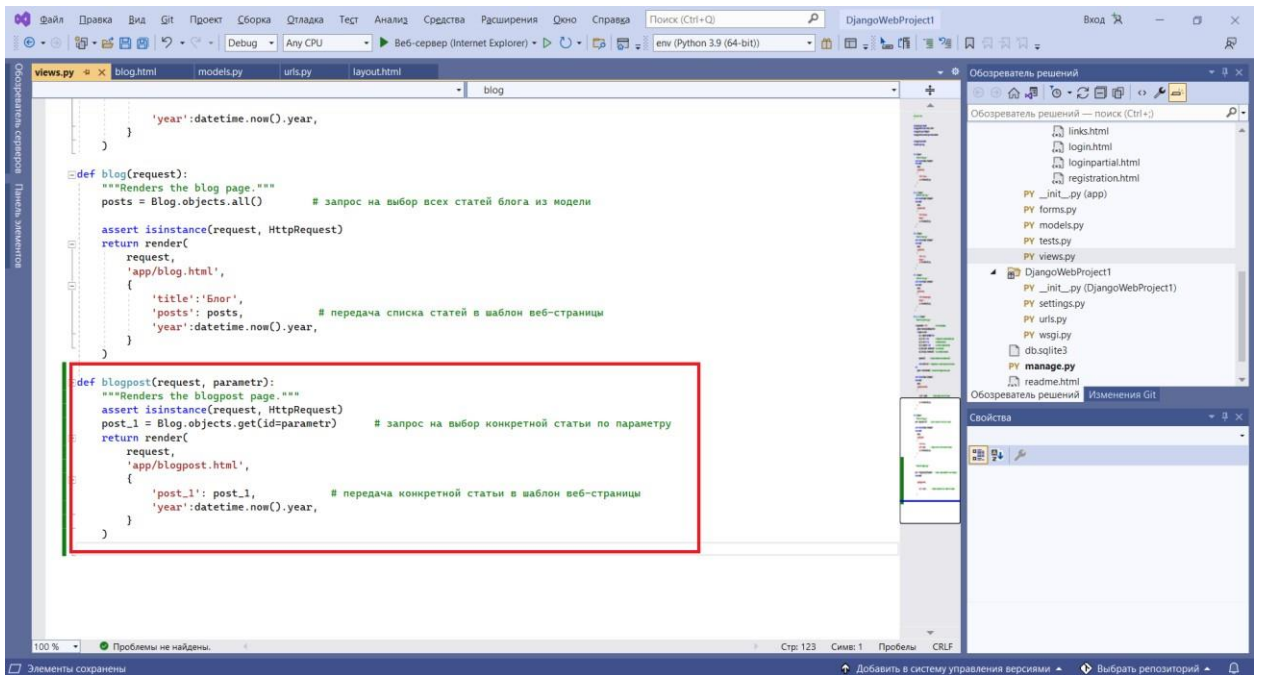


Рисунок 9 –Код метода действия контроллера для страницы поста
 d. Добавить страницу поста (см. рисунок 10).

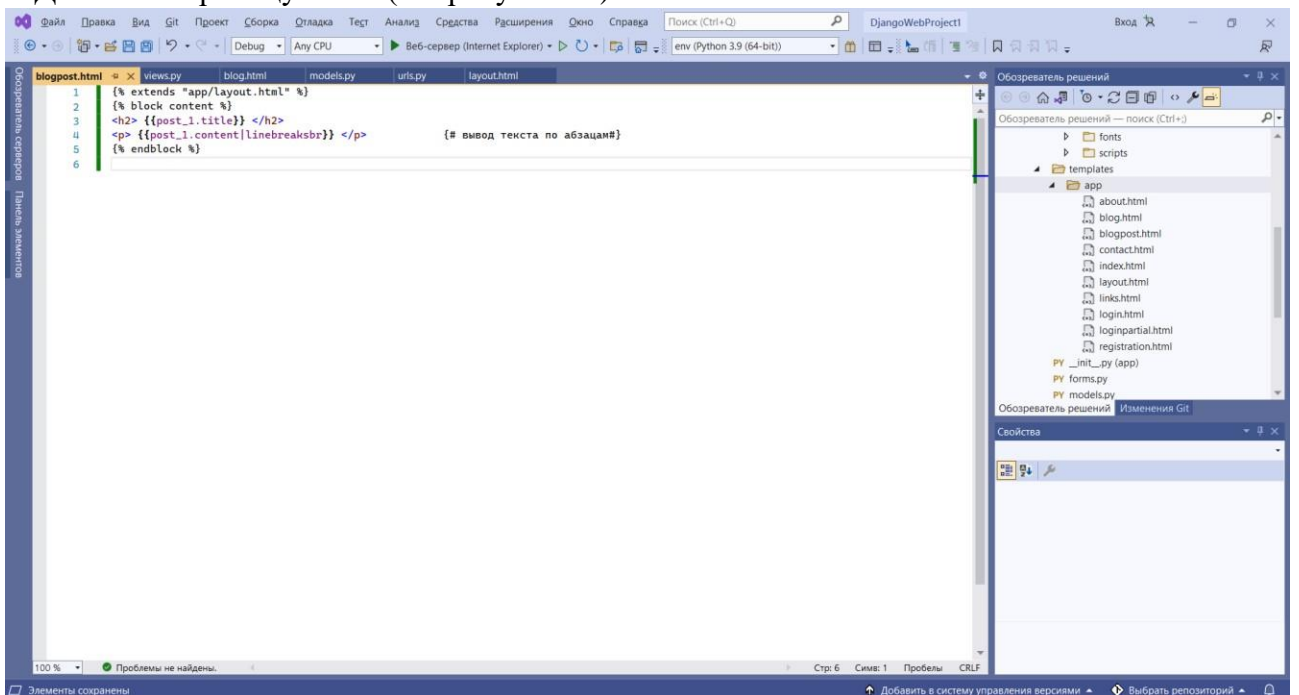


Рисунок 10 – Код веб-страницы `blogpost.html`
 e. Проверить отображение статей блога на сайте (результат на рисунке 11).

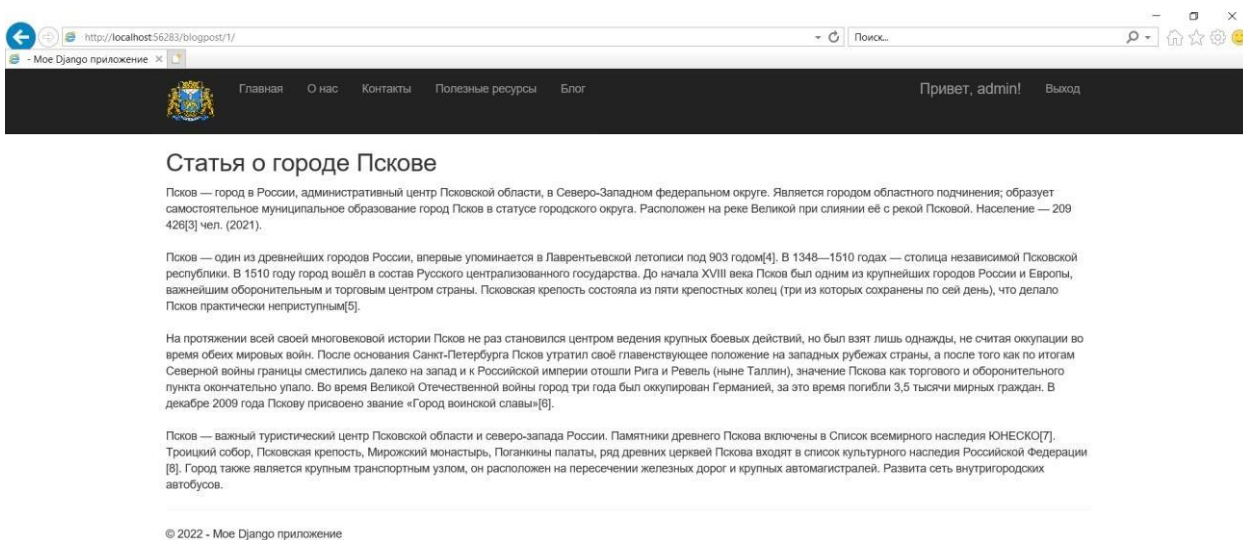


Рисунок 11 – Отображение статьи блога

3. Добавить в файл **blogpost.html** после вывода текста поста ссылку **Назад** на страницу **blog.html** (см. рисунок 12).

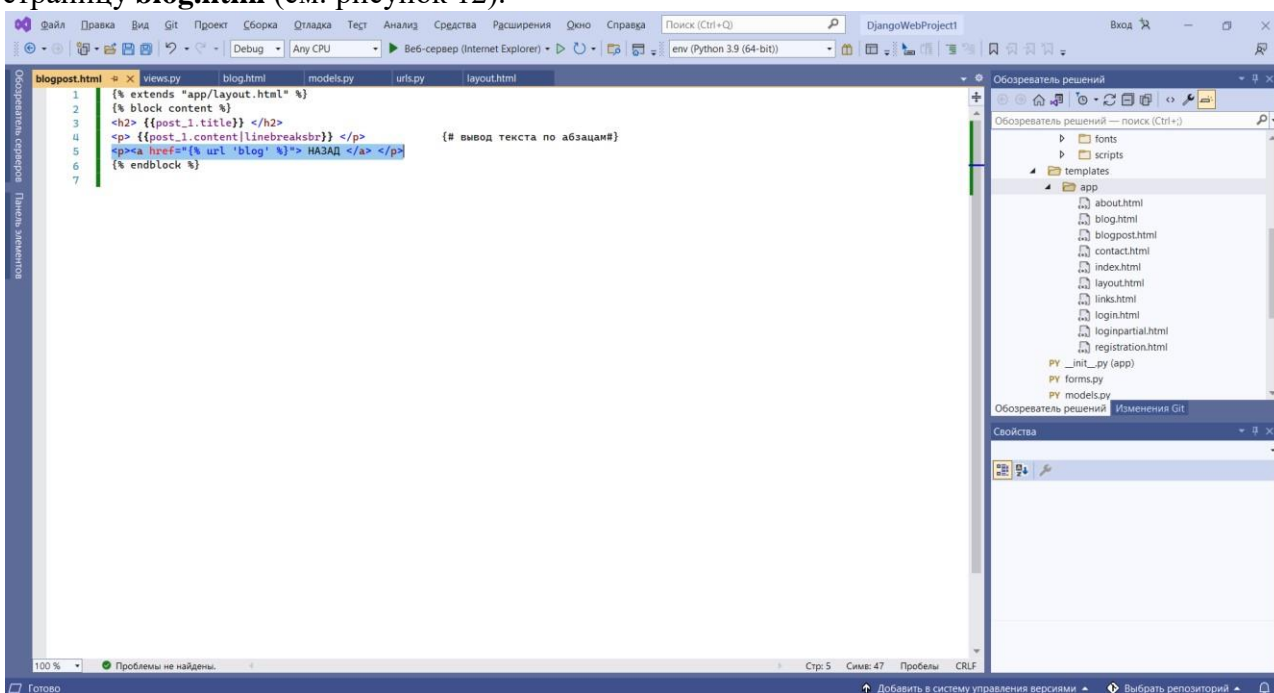


Рисунок 12 – Добавление на страницу поста ссылки Назад

4. Войти в административный раздел Django под логином и паролем администратора сайта. Изменить содержание статей в соответствии с выбранной темой. Перейти на сайт по ссылке **ОТКРЫТЬ САЙТ**. Посмотреть корректность отображения статей блога на сайте.

5. Применить индивидуальное оформление для элементов веб-страниц сайта. На основе материалов темы 2 необходимо:

а) В файле **layout.html** изменить строку подключения **bootstrap.min.css** на **bootstrap.css**:

```
<link rel="stylesheet" type="text/css" href="{% static 'app/content/bootstrap.css' %}" />
```

б) В файле **bootstrap.css** изменить стили, например:

для меню:

```
/*цвет фона навигационной панели – меню*/
```

```

.navbar-inverse {
    background-color: #5bc0de;
    border-color: #080808;
}
/*цвет шрифта меню*/
.navbar-inverse .navbar-nav > li > a {
    color: #ff0000;
}
/*стиль пунктов меню при наведении мыши*/
.navbar-inverse .navbar-nav > li > a:hover,
.navbar-inverse .navbar-nav > li > a:focus {
    color: #00ff21;
    background-color: orangered;
    border-radius:20%;
}

```

Лабораторная работа 9. Добавление комментариев к статьям блога

1. необходимо:

a. Открыть, созданный ранее, веб-проект Django в Visual Studio. В нём открыть файл **models.py**.

b. В начале файла **models.py** после уже имеющихся строк импорта должна быть добавлена строка импорта модели **User** (в рамках выполнения задания в самостоятельной работе по теме 7):

```
from django.contrib.auth.models import User
```

c. В файл **models.py** добавить код класса модели комментариев – **Comment**, содержащий:

- Поля модели для текста комментария, даты, имени пользователя и идентификатора статьи для комментария (см. рисунок 1).
- Метод **__str__** (см. рисунок 1).
- Метаданные – **class Meta** (см. рисунок 1).

d. В конец файла **models.py** после объявления модели (class Comment) необходимо добавить строку регистрации модели комментариев в административном разделе (см. рисунок 1):

```
admin.site.register(Comment)
```

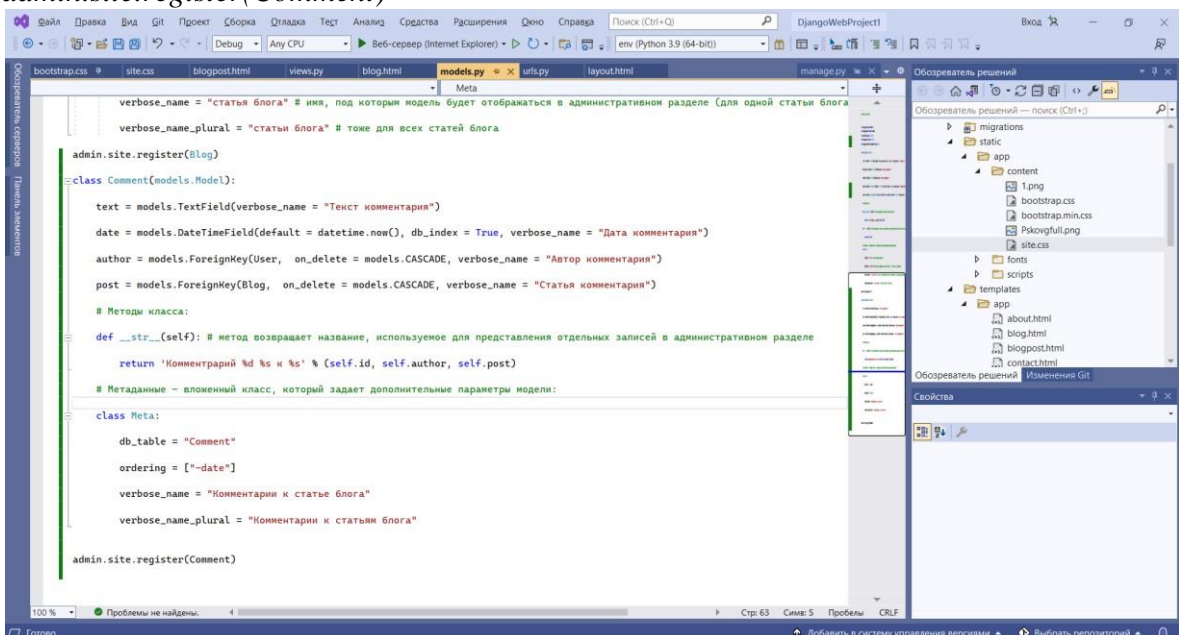


Рисунок 1 – Добавление в веб-проект модели комментариев к статье блога

е. Обновить базу данных сайта (см. рисунок 2):

- В меню **Проект** выбрать команду **Миграция сборки Django**
- В меню **Проект** выбрать команду **Миграция Django...**

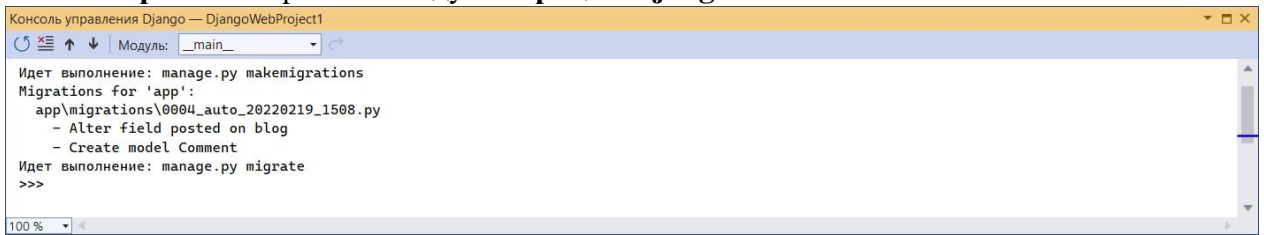


Рисунок 2 – Обновление базы данных (создание таблицы Comments)

2. На основе материалов в подразделе **Теория для практического занятия** необходимо:

а. Добавить в начало файла **forms.py** строк импорта базового класса и класса модели Comment (см. рисунок 3).

б. Добавить в конец файла **forms.py** код для формы ввода комментария (см. рисунок 3).

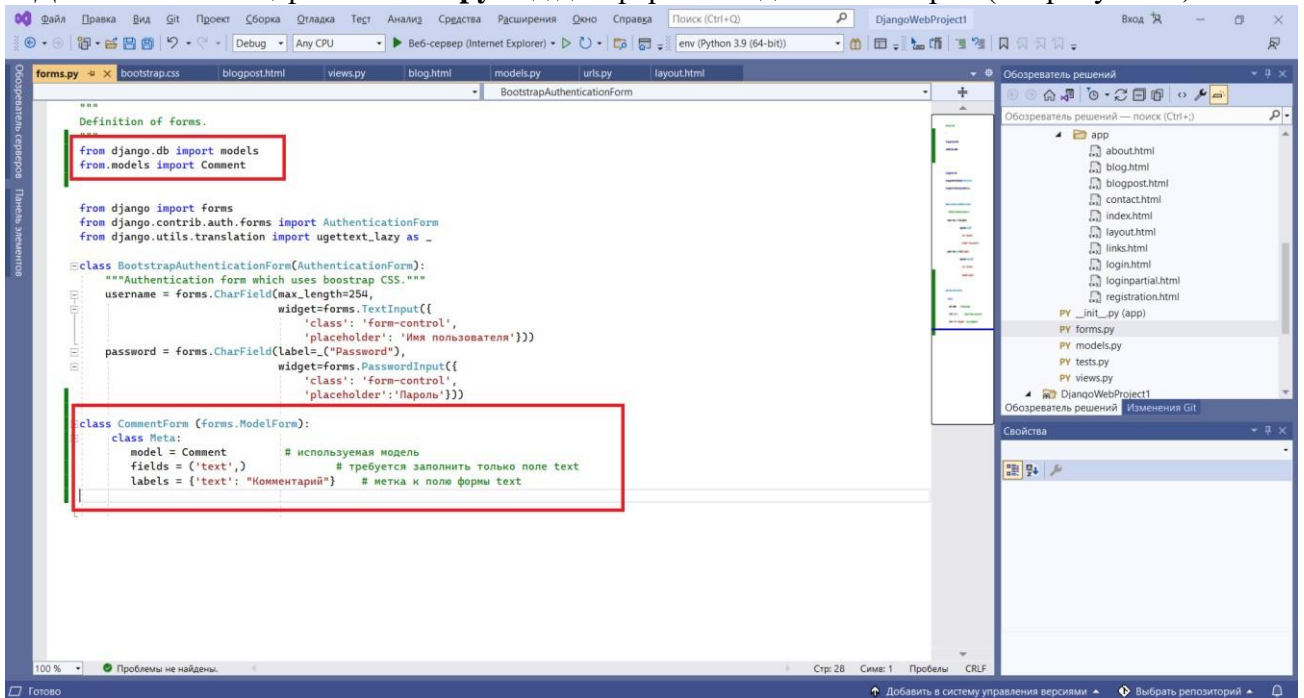


Рисунок 3 – Добавление в проект формы для ввода комментария

с. Отобразить форму ввода комментария в шаблоне веб-страницы **blogpost.html** (см. рисунок 4).

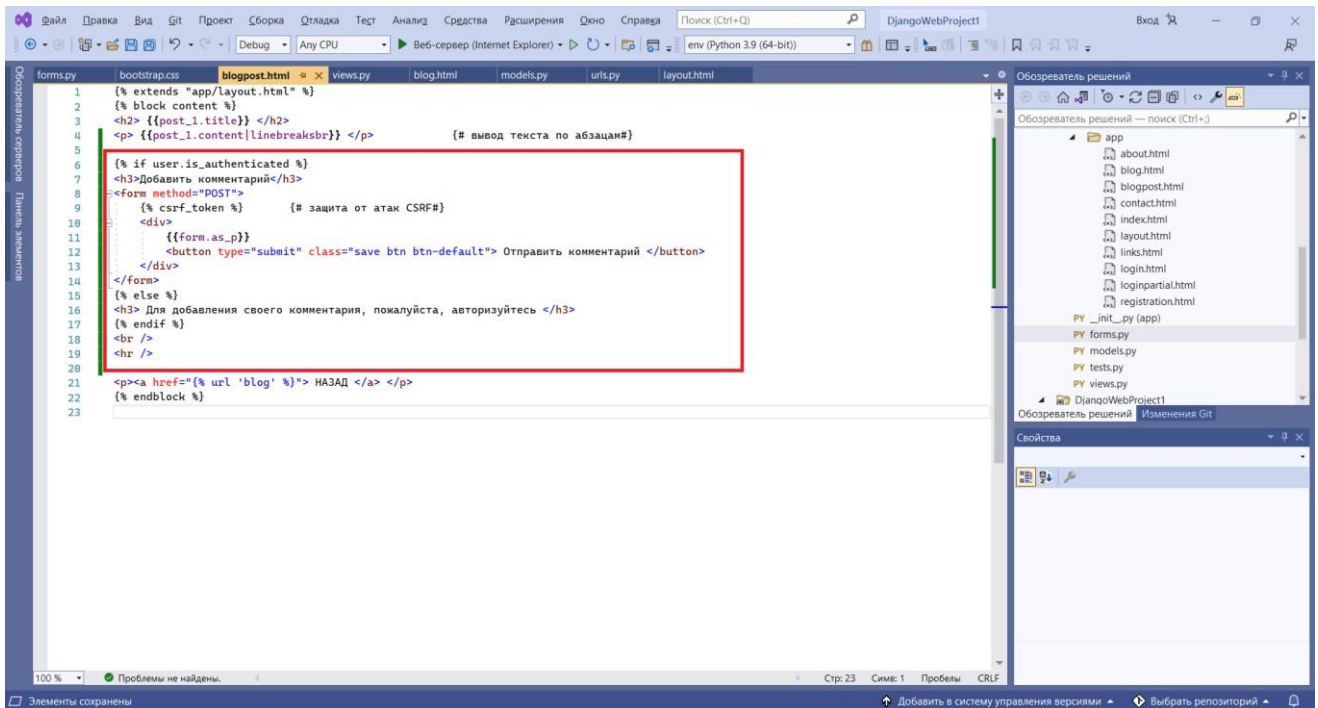


Рисунок 4 – Добавление формы ввода комментария на веб-страницу

d. В файл **blogpost.html** после кода отображения формы ввода комментария добавить код вывода всех комментариев к статье блога (см. рисунок 5).

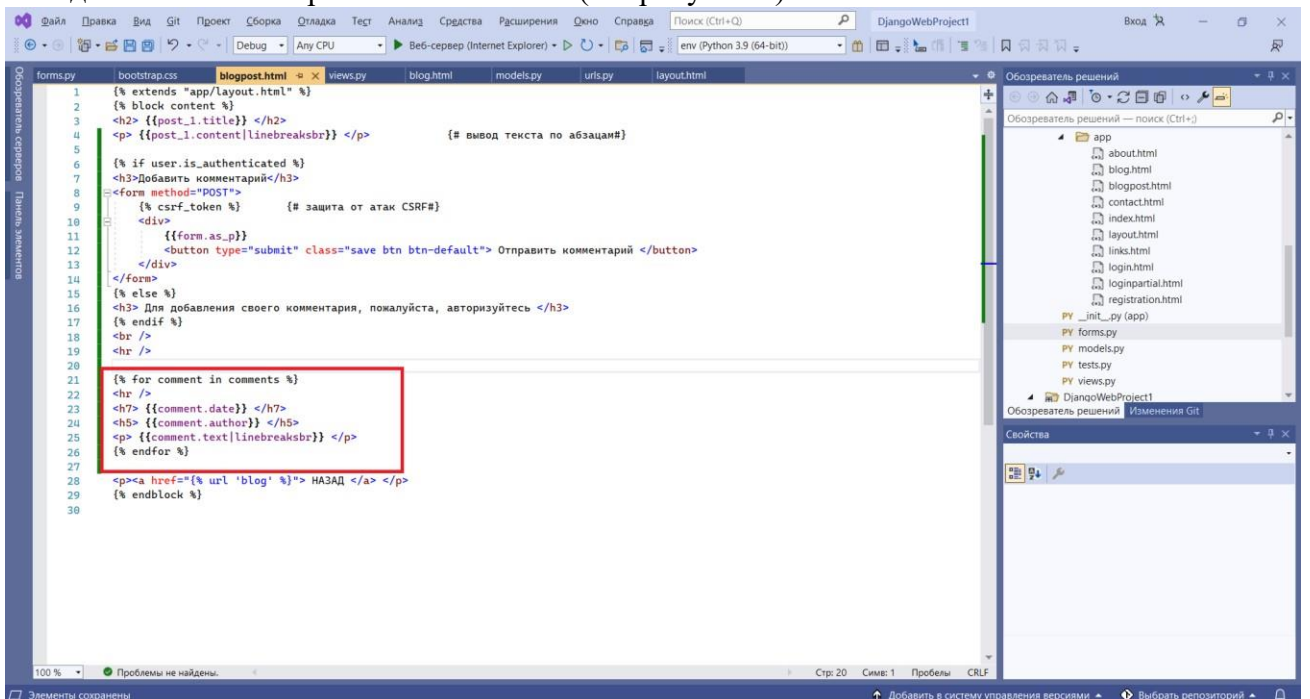


Рисунок 5 – Вывод всех комментариев к статье блога

e. Добавить в **views.py** строки импорта модели комментариев и формы для ввода комментариев (см. рисунок 6)

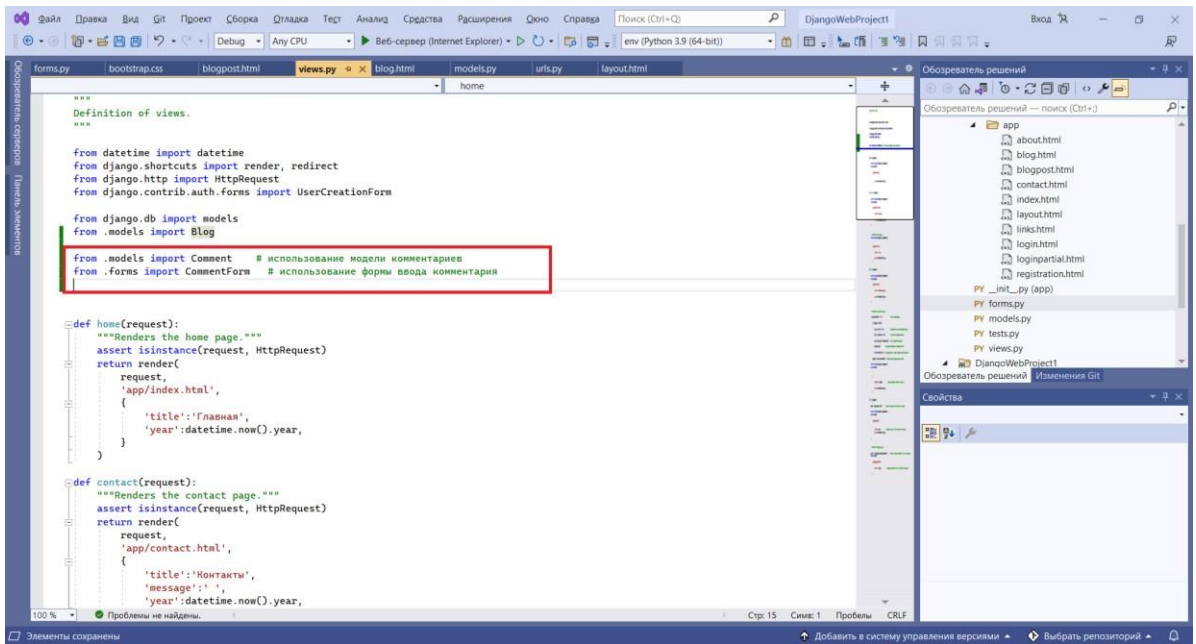


Рисунок 6 – Импорт модели комментариев и формы ввода комментариев

f. Добавить в **views.py** запрос на выбор всех комментариев статьи блога (см. рисунок 7).

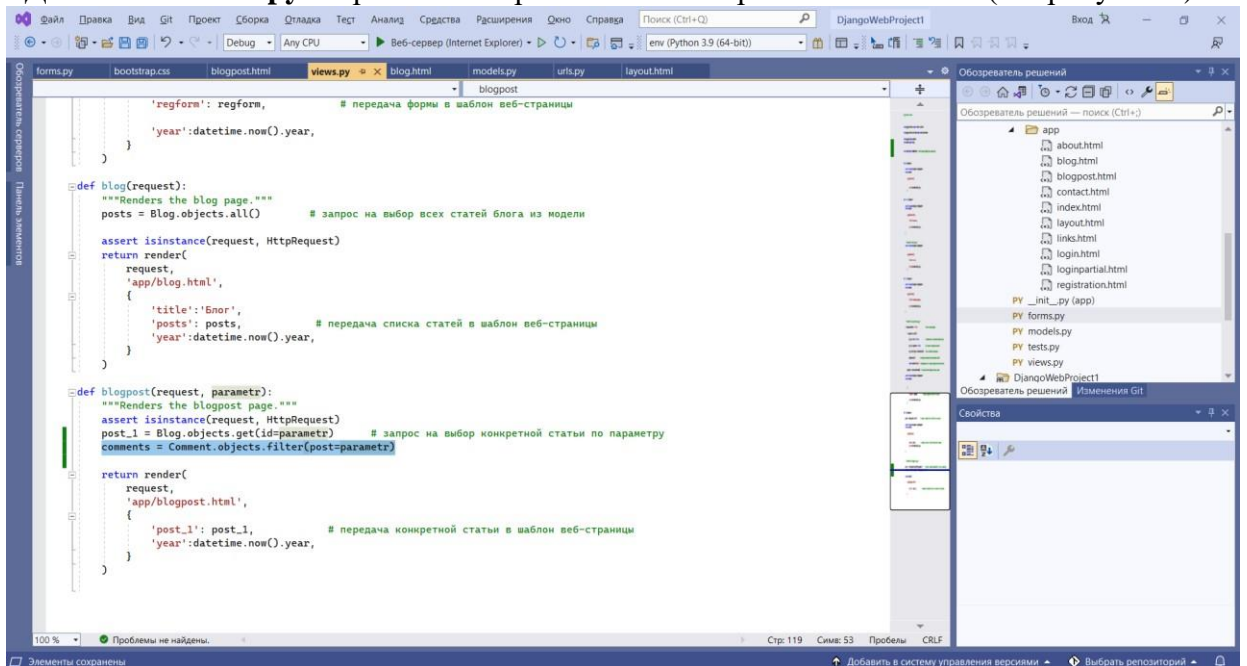


Рисунок 7 – Запрос на выбор всех комментариев статьи блога

g. Добавить код работы с формой в файл **views.py** в код метода контроллера *def blogpost* (*request, parametr*): (см. рисунок 8).

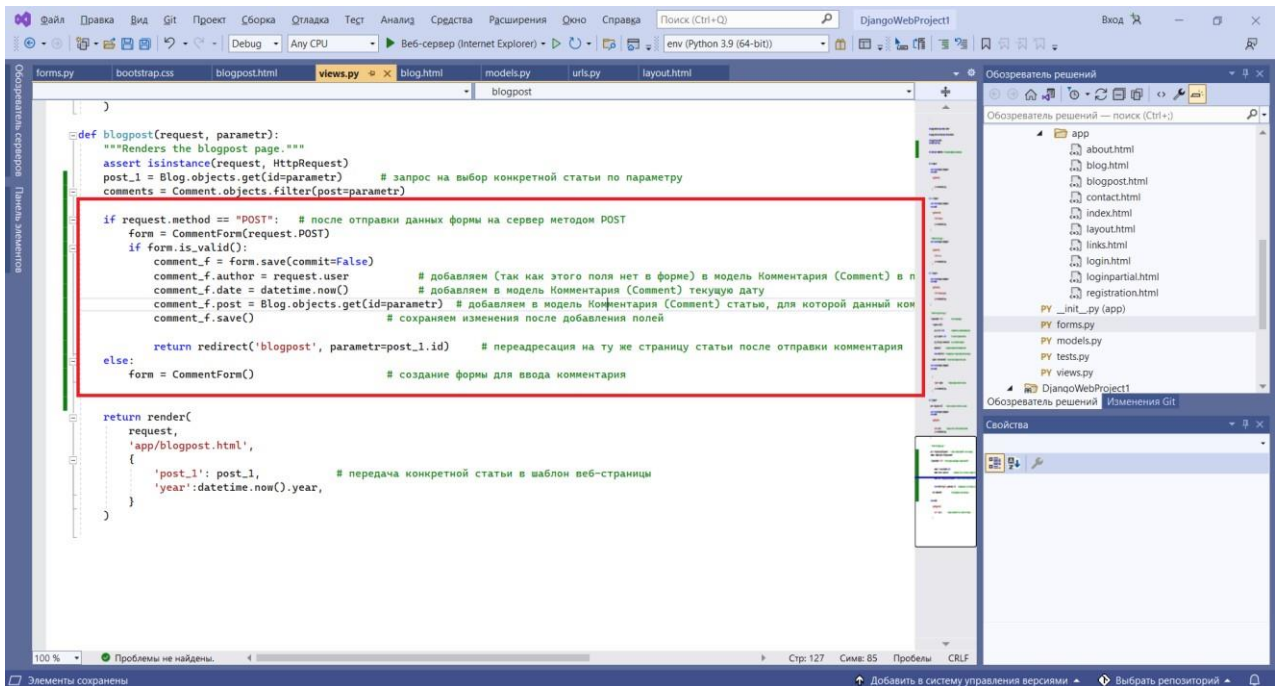


Рисунок 8 – Код работы с формой

h. В файле **views.py** в код метода контроллера *def blogpost (request, parametr):* в методе **render** добавить передачу объектов формы добавления комментария и всех комментариев к конкретной статье в шаблон веб-страницы **blogpost.html** (см. рисунок 9).

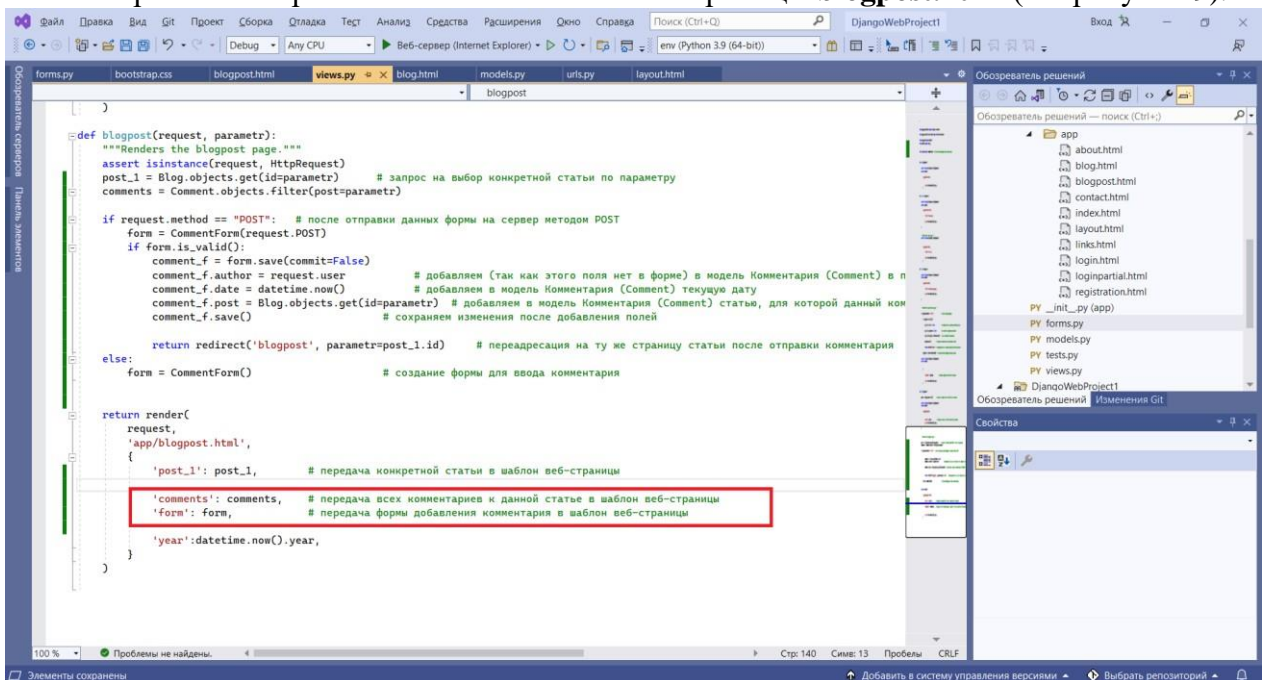


Рисунок 9 – Передача объектов формы добавления комментария и всех комментариев к конкретной статье в шаблон веб-страницы **blogpost.html**

3. Задайте в файле **site.css**, например, цвет шрифта для заголовка **h5**, которым отображается автор комментария:

```

h5 {
    color:aqua;
}

```

4. Запустить проект на выполнение, авторизоваться, перейти на страницу блога, перейти на страницу поста, добавить комментарий к посту (см. рисунок 10).

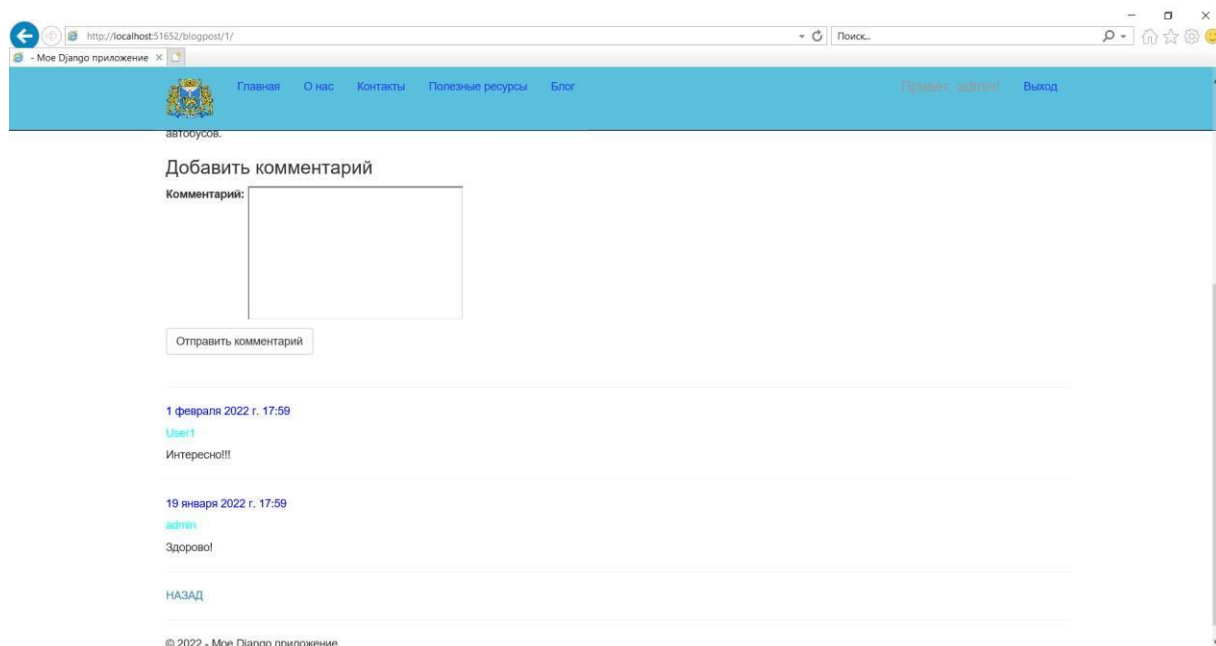


Рисунок 10 – Результат добавления комментария к статье блога

Добавление в проект раздела сайта с изображениями и видео по тематике сайта

1. необходимо:

а. Добавить в модель записи блога поле для картинки (см. рисунок 1):

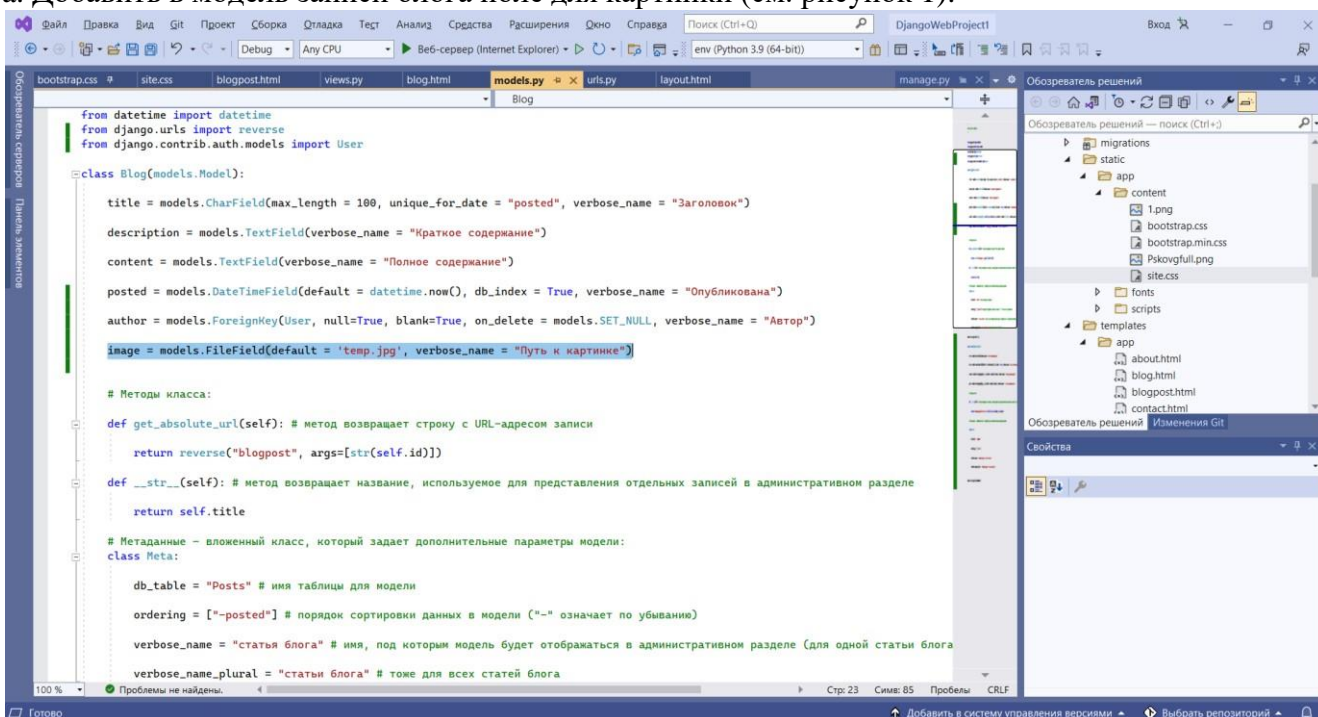


Рисунок 1 – Добавление в модель записи блога поле для картинки

б. Выполните миграцию для обновления базы данных:

- В меню **Проект** выбрать команду **Миграция сборки Django**
- В меню **Проект** выбрать команду **Миграция Django...**

с. В файл **settings.py** добавить определение переменных для загрузки графических файлов в папку **media** (см. рисунок 2).

д. В том же файле в список **context_processors** (в переменной **TEMPLATES** в поле **OPTIONS**) добавить строку для подключения обработчика загрузки файлов (см. рисунок 3).

е. В файл **urls.py** добавить импорт функций для настройки доступа к загруженным файлам (см. рисунок 4).

f. В конец файла *urls.py* добавить строки шаблонов URL для загрузки файлов в папку *media*, вложенной в папку проекта (см. рисунок 5).

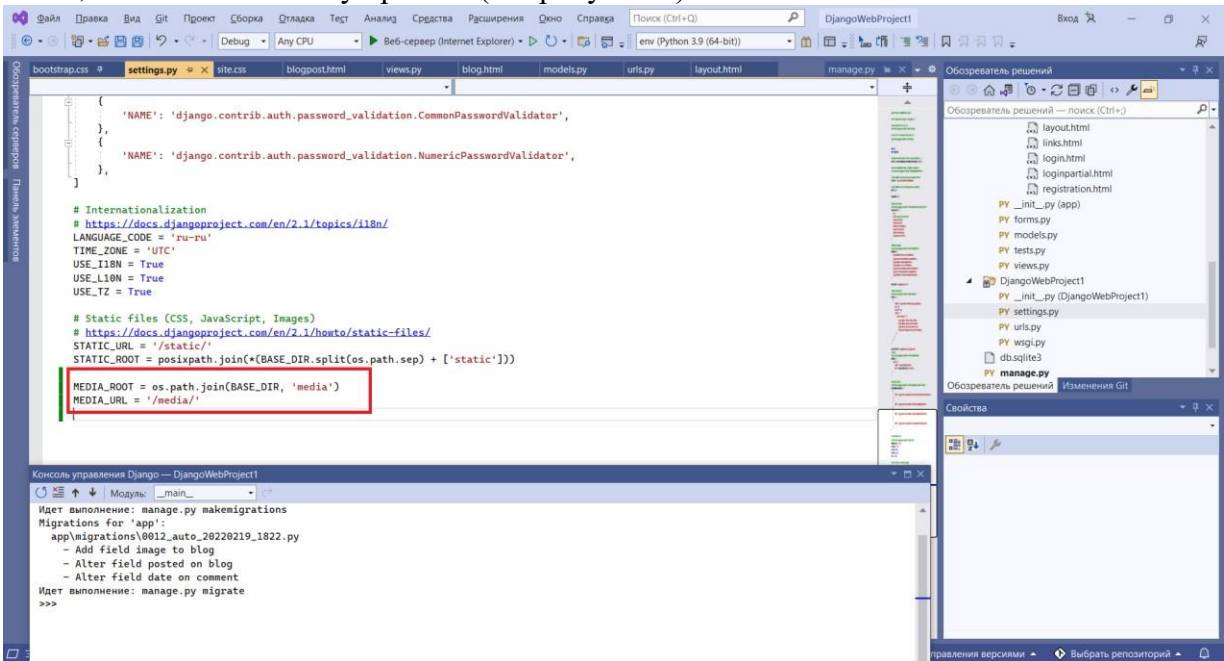


Рисунок 2 – Добавление в файл *settings.py* определение переменных для загрузки файлов в папку *media*

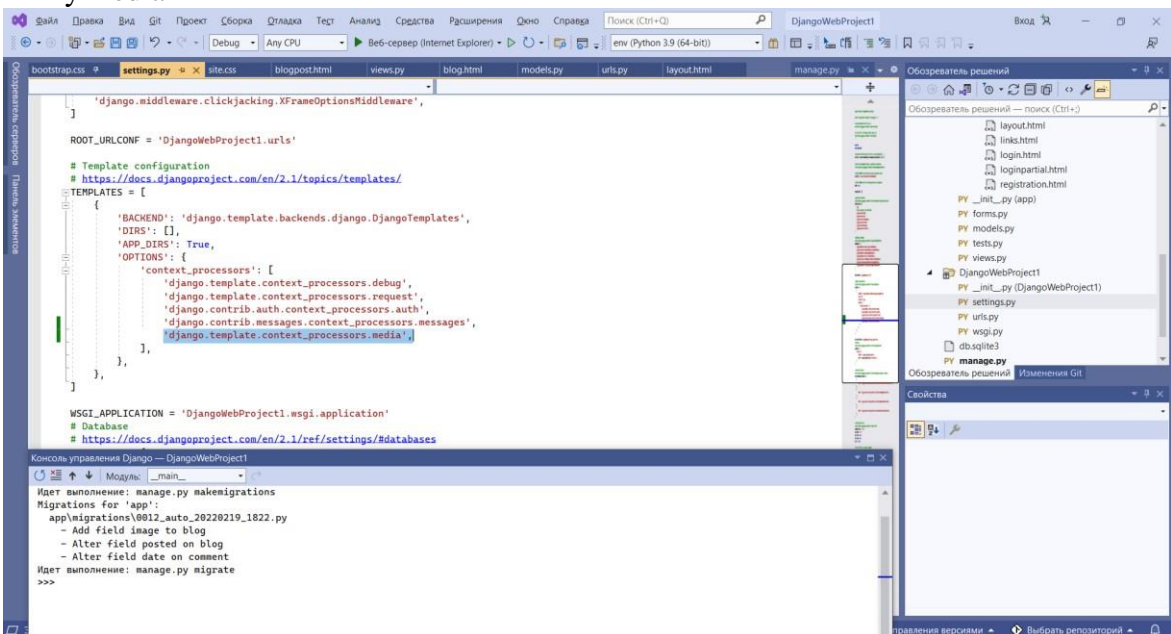


Рисунок 3 – Добавление строки для подключения обработчика загрузки файлов

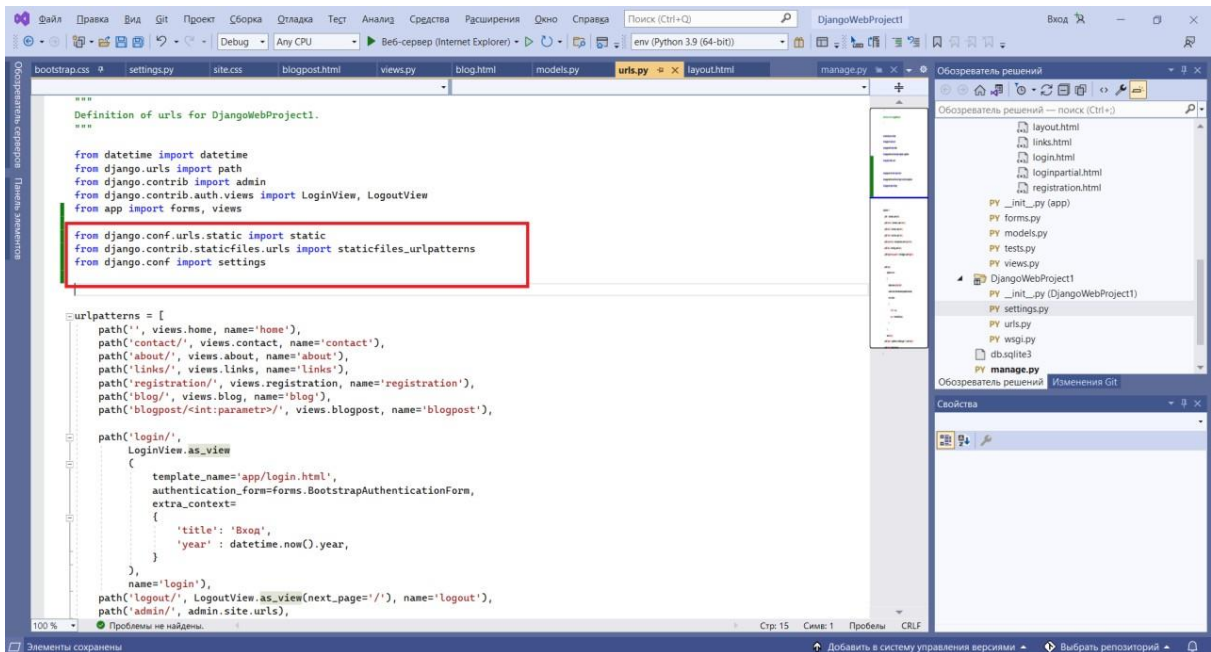


Рисунок 4 – Добавление в файл urls.py строк импорта функций для настройки доступа к загруженным файлам

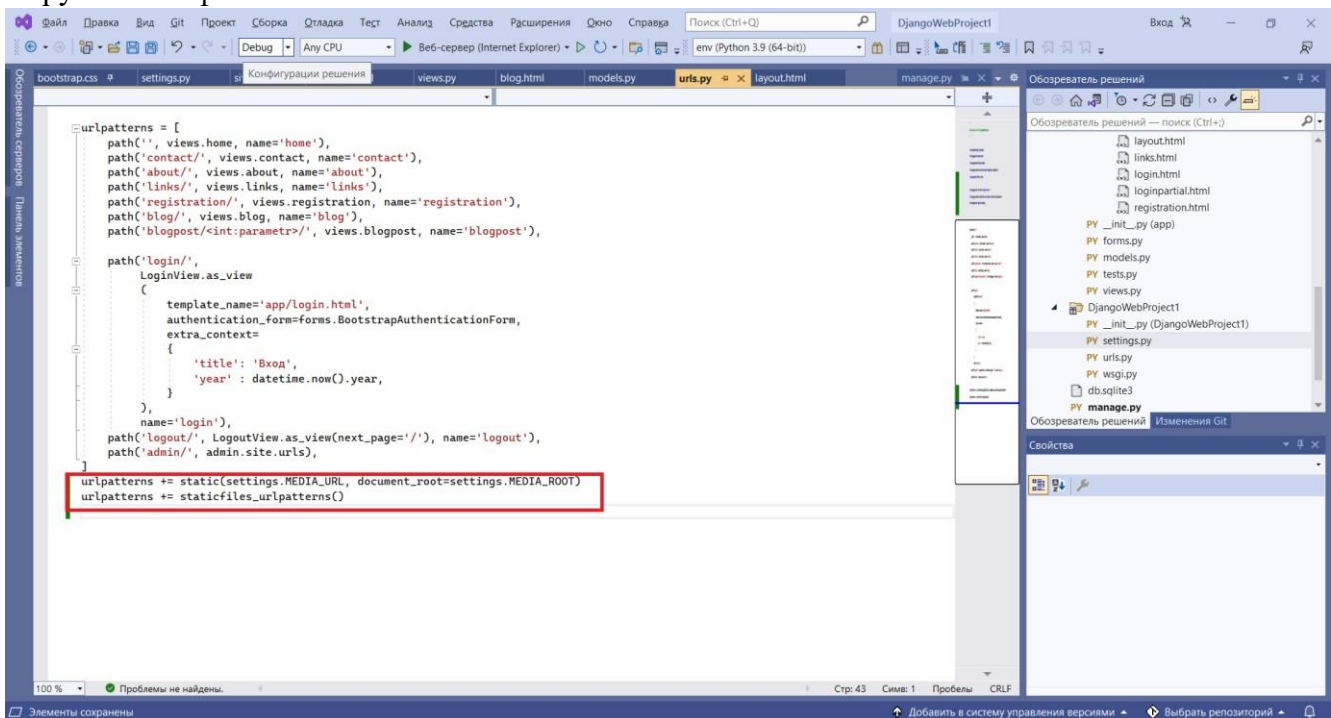


Рисунок 5 – Добавление строк шаблонов URL для загрузки файлов в папку media, вложенной в папку проекта

2. На основе материалов лекции 11 и из теории в подразделе **Теория для практического занятия** следует, что для реализации добавления статей блога администратором на отдельной веб-странице сайта, необходимо:

а. Добавить в начало файла **forms.py** строку импорта модели блога (см. рисунок 6):

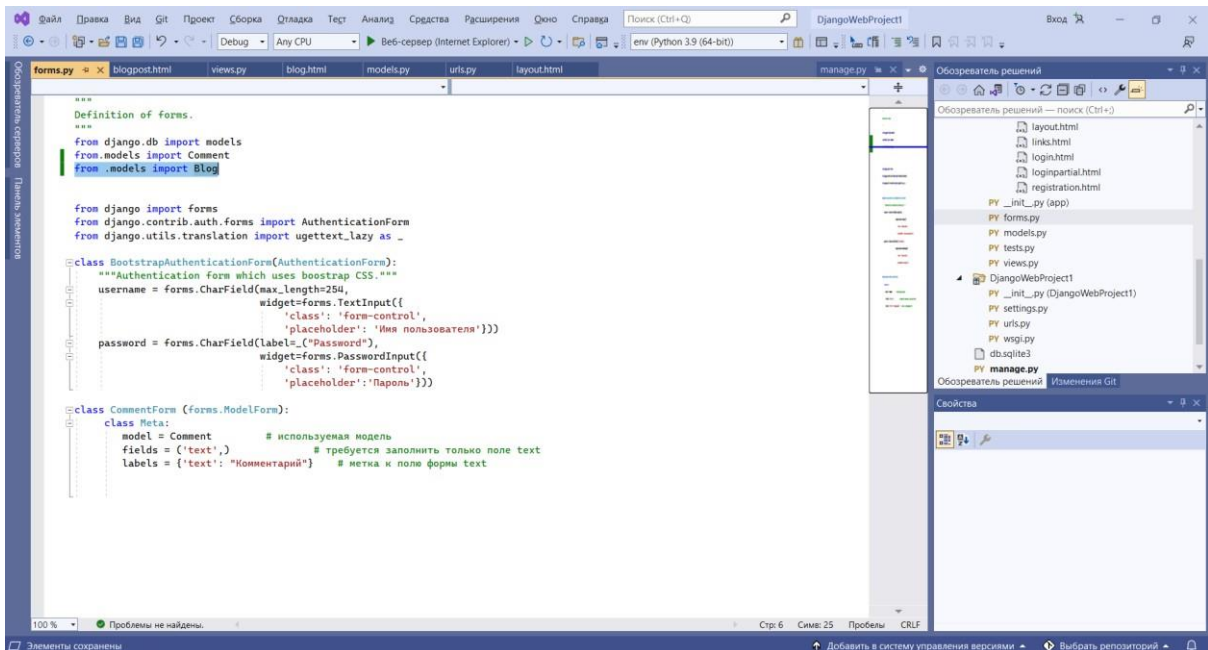


Рисунок 6 – Добавление в начало файла forms.py строку импорта модели блога
 в. Добавить в конец файла **forms.py** код формы ввода статьи блога. Форма должна содержать поля для ввода данных статьи блога, например, для ввода заголовка, краткого содержания, полного содержания статьи, выбора файла картинки (см. рисунок 7).

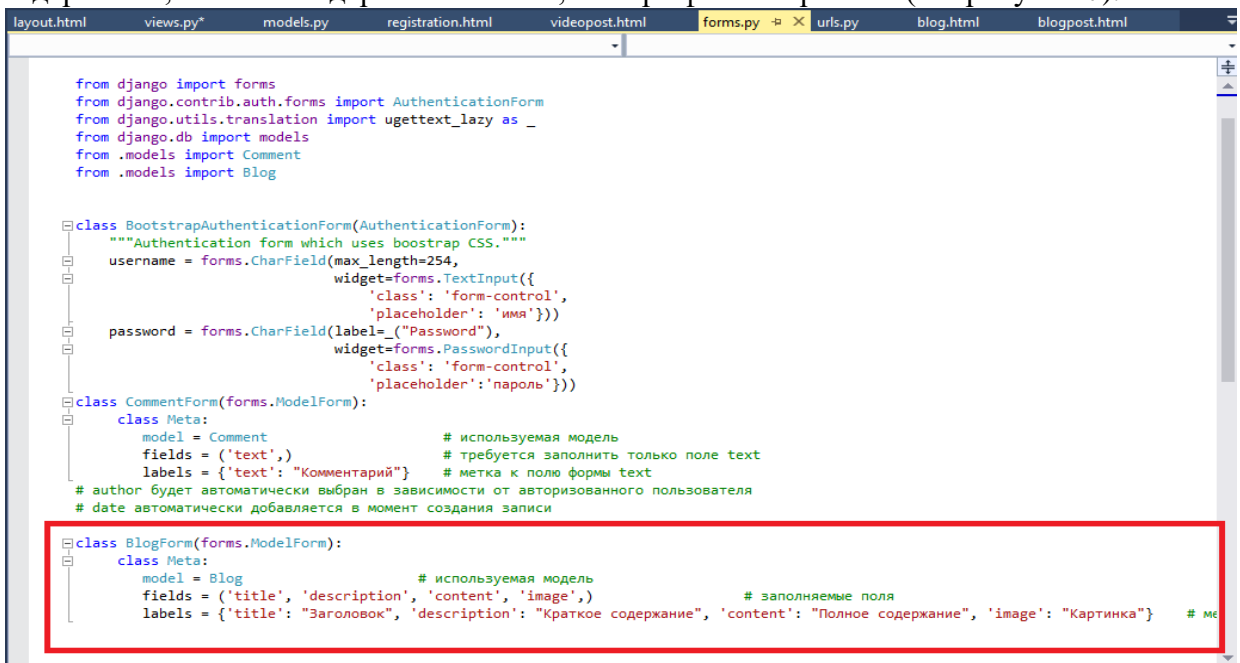


Рисунок 7 – Добавление в forms.py кода формы ввода статьи блога
 с. Добавить в проект шаблон для html-файла для ввода статьи блога (например, *newpost.html*).
 d. Добавить код для отображения формы ввода статьи блога в **newpost.html**. В тег формы добавить атрибут *enctype="multipart/form-data"* (см. рисунок 8).
 e. Добавить в начало файла **views.py** код импорта формы для ввода статьи блога.
 f. В файл **views.py** добавить код метода действия контроллера *def newpost(request):* (см. рисунок 9).
 г. Добавить строку шаблона URL в файл **urls.py** для страницы добавления статьи блога (см. рисунок 10).

```

newpost.html  ↳ layout.html  views.py*  models.py  registration.html  videopost.html  forms.py  urls.py  blog.html
{% extends "app/layout.html" %}

{% block content %}

<h3>Добавить статью</h3>
<form method="POST" enctype="multipart/form-data"> {% csrf_token %}  {# обязательная защита от атак CSRF#}
<div>
    {{ blogform.as_p }}
    <button type="submit" class="save btn btn-default"> Сохранить статью </button>
</div>
</form>

{% endblock %}

```

Рисунок 8 – Добавление кода для отображения формы ввода статьи блога в newpost.html

```

layout.html  views.py*  models.py  registration.html  videopost.html  forms.py  urls.py  blog.html  blogpost.html
newpost
    'year':datetime.now().year,
    )
}

def newpost(request):
    """Renders the newpost page."""
    assert isinstance(request, HttpRequest)

    if request.method == "POST":
        blogform = BlogForm(request.POST, request.FILES)
        if blogform.is_valid():
            blog_f = blogform.save(commit=False)
            blog_f.posted = datetime.now()
            blog_f.autor = request.user
            blog_f.save()
            return redirect('blog')
        else:
            blogform = BlogForm()

    return render(
        request,
        'app/newpost.html',
        {
            'blogform': blogform,
            'title': 'Добавить статью блога',
            'year':datetime.now().year,
        }
    )

```

Рисунок 9 – Добавление кода метода действия контроллера def newpost(request):

```

from django.contrib import admin
from django.contrib.auth.views import LoginView, LogoutView
from app import forms, views

from django.conf.urls.static import static
from django.contrib.staticfiles.urls import staticfiles_urlpatterns
from django.conf import settings

urlpatterns = [
    path('', views.home, name='home'),
    path('contact/', views.contact, name='contact'),
    path('about/', views.about, name='about'),
    path('links/', views.links, name='links'),
    path('registration/', views.registration, name='registration'),
    path('blog/', views.blog, name='blog'),
    path('blogpost/<int:parameter>', views.blogpost, name='blogpost'),
    path('newpost/', views.newpost, name='newpost'),

    path('login/',
        LoginView.as_view(
            template_name='app/login.html',
            authentication_form=forms.BootstrapAuthenticationForm,
            extra_context={
                'title': 'Вход',
                'year': datetime.now().year,
            },
        ),
        name='login'),
    path('logout/', LogoutView.as_view(next_page='/', name='logout'),
    path('admin/', admin.site.urls),

    urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
    urlpatterns += staticfiles_urlpatterns()

```

Рисунок 10 – Добавление строки шаблона URL в файл urls.py для страницы добавления статьи блога

h. Добавить пункт в меню сайта для страницы добавления статьи блога, доступный только для администратора – метод *is_superuser* (см. рисунок 11).

i. В файл **blogpost.html** добавить вывод картинки записи блога перед содержанием статьи (см. рисунок 12).

j. Запустить проект на выполнение. Войти на сайт под администратором. Добавить статью блога с картинкой, добавить к ней комментарий. Проверить отображение новой статьи и комментарий.

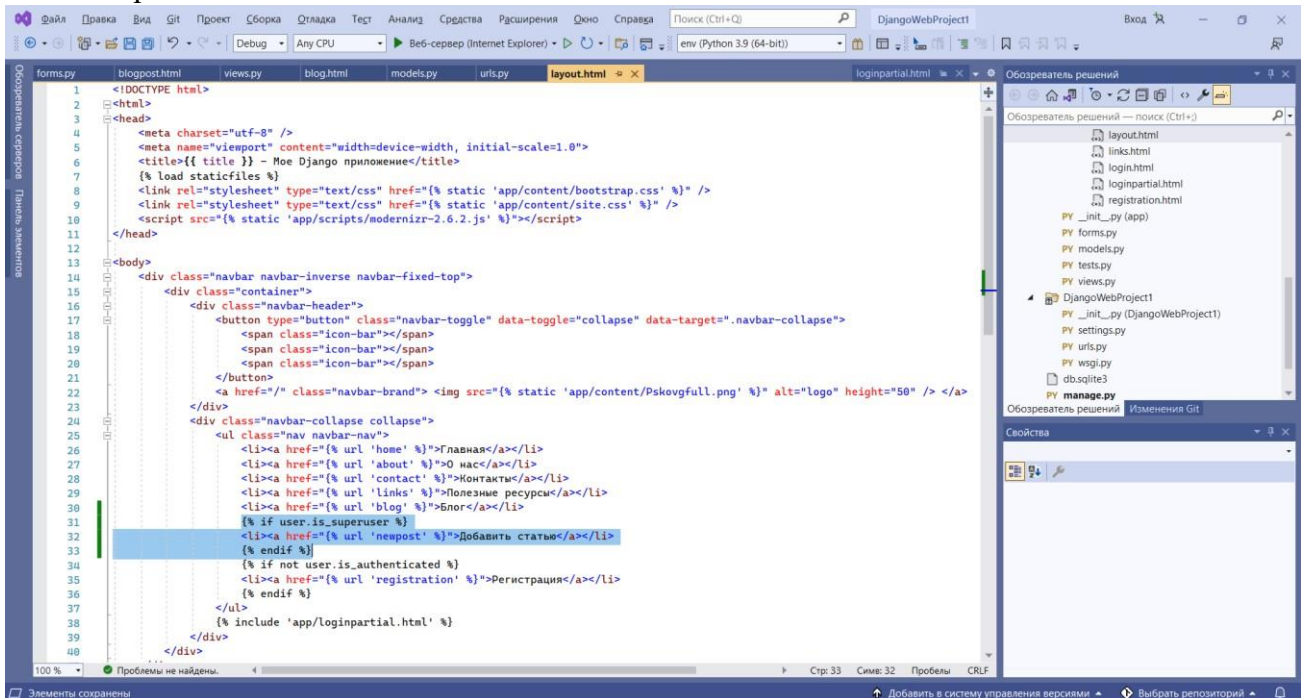


Рисунок 11 – Добавление пункта в меню сайта для страницы добавления статьи блога

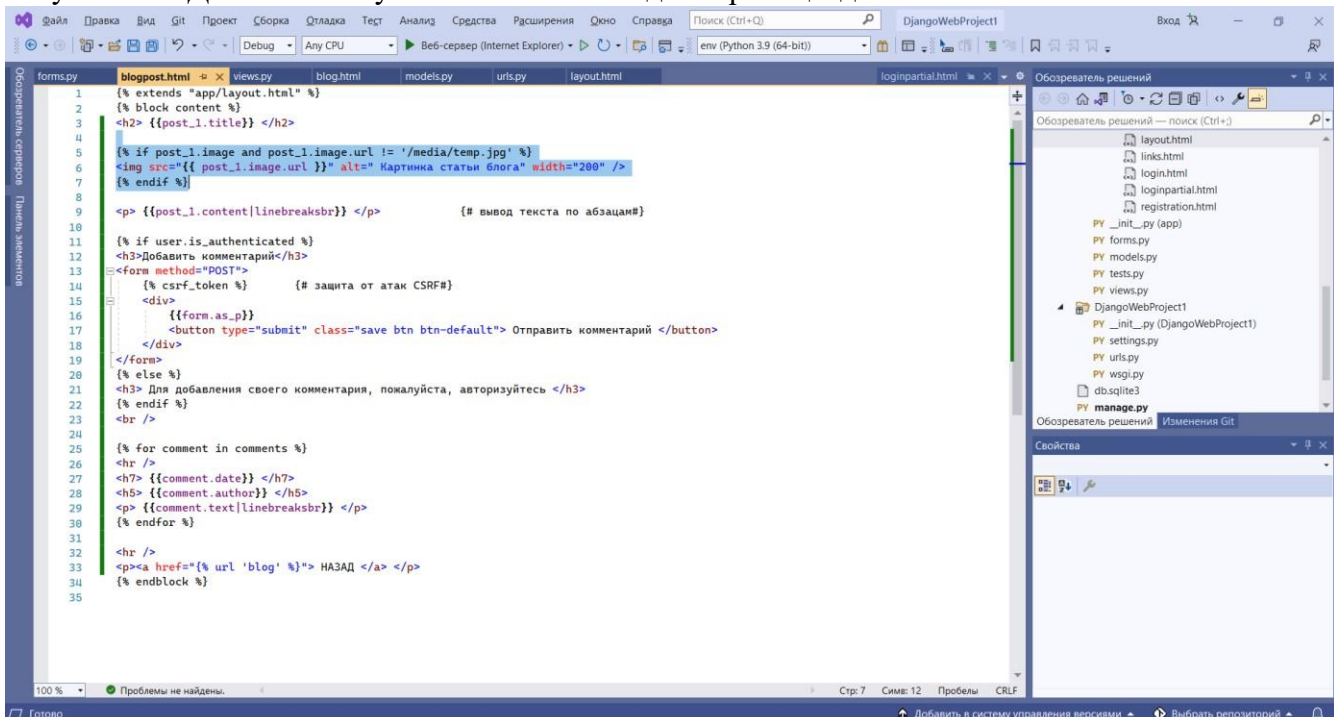
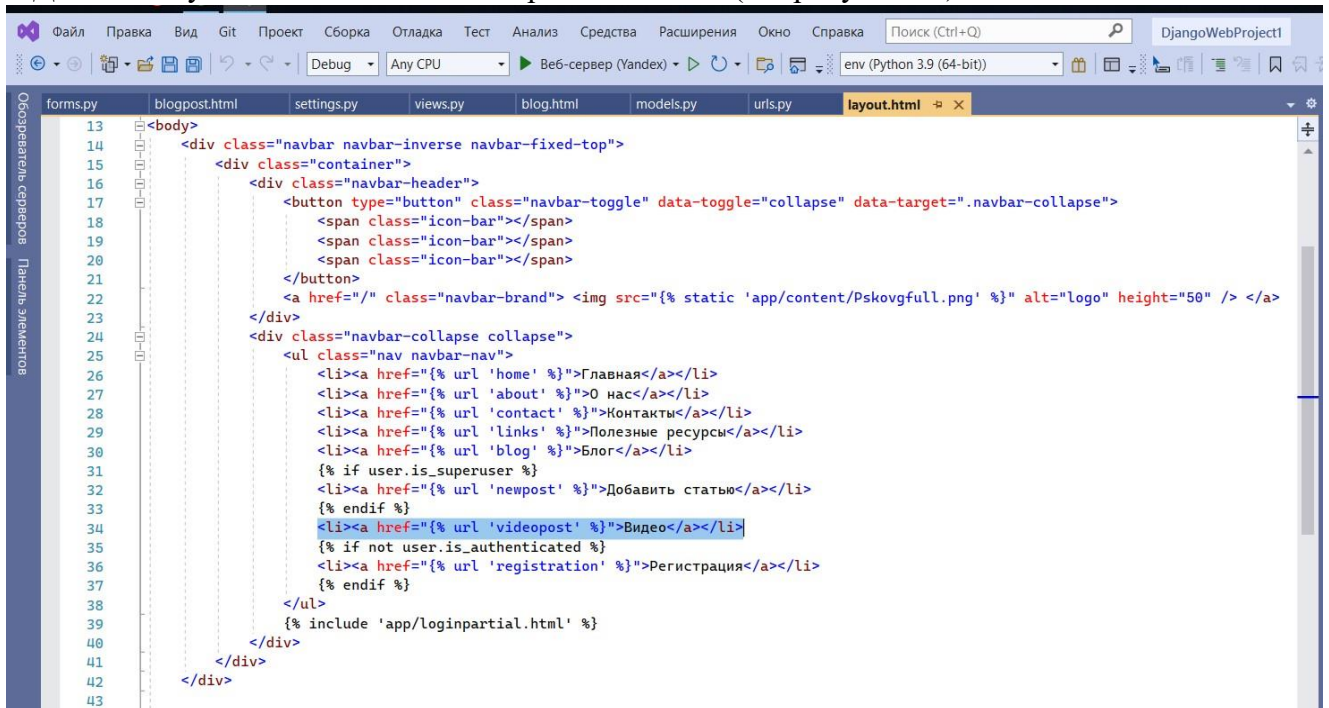


Рисунок 12 – Добавление вывода картинки записи блога перед содержанием статьи

3. На основе материалов лекции 11 и из теории в подразделе **Теория для практического занятия** следует, что для реализации на отдельной веб-странице сайта (например,

«Видео») отображение и воспроизведение двух видео-фрагментов формата .mp4 и длительностью не более 1 минуты необходимо:

- a. Первоначально подготовить (или найти) 2 видео-фрагмента по тематике сайта формата .mp4, длительностью не более 1 минуты и картинки к ним для заставки (poster).
- b. Скопировать подготовленные 2 видео-файла и картинки к ним (poster) в папку **media** веб-проекта.
- c. Добавить пункт в меню сайта для страницы **Видео** (см. рисунок 13):



```
13 </body>
14 <div class="navbar navbar-inverse navbar-fixed-top">
15 <div class="container">
16 <div class="navbar-header">
17 <button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-collapse">
18 <span class="icon-bar"></span>
19 <span class="icon-bar"></span>
20 <span class="icon-bar"></span>
21 </button>
22 <a href="/" class="navbar-brand"> </a>
23 </div>
24 <div class="navbar-collapse collapse">
25 <ul class="nav navbar-nav">
26 <li><a href="{% url 'home' %}">Главная</a></li>
27 <li><a href="{% url 'about' %}">О нас</a></li>
28 <li><a href="{% url 'contact' %}">Контакты</a></li>
29 <li><a href="{% url 'links' %}">Полезные ресурсы</a></li>
30 <li><a href="{% url 'blog' %}">Блог</a></li>
31 {% if user.is_superuser %}
32 <li><a href="{% url 'newpost' %}">Добавить статью</a></li>
33 {% endif %}
34 <li><a href="{% url 'videopost' %}">Видео</a></li>
35 {% if not user.is_authenticated %}
36 <li><a href="{% url 'registration' %}">Регистрация</a></li>
37 {% endif %}
38 </ul>
39 {% include 'app/loginpartial.html' %}
40 </div>
41 </div>
42 </div>
43
```

Рисунок 13 – Добавление пункта в меню сайта для страницы Видео

- d. Добавить строку шаблона URL в файл **urls.py** для страницы **Видео** (см. рисунок 14).
- e. Добавить в файл **views.py** код метода действия контроллера *def videopost(request)*: (по аналогии с методом about).
- f. Добавить в проект шаблон для html-файла для видео-контента (например, *videopost.html*).
- g. Добавить код для отображения и воспроизведения видео-роликов в **videopost.html** (см. рисунок 15).
- h. Запустить проект на выполнение. Проверить отображение и воспроизведение видео-роликов на странице сайта.

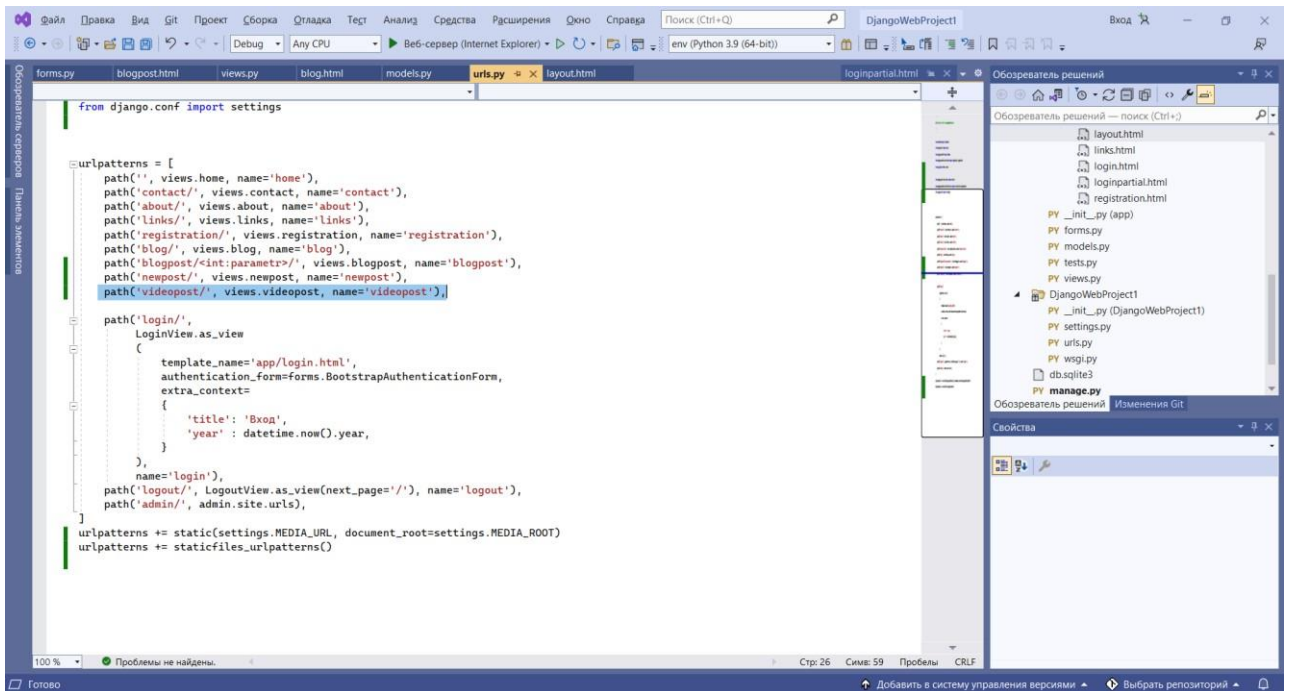


Рисунок 14 – Добавление строки шаблона URL в файл urls.py для страницы Видео

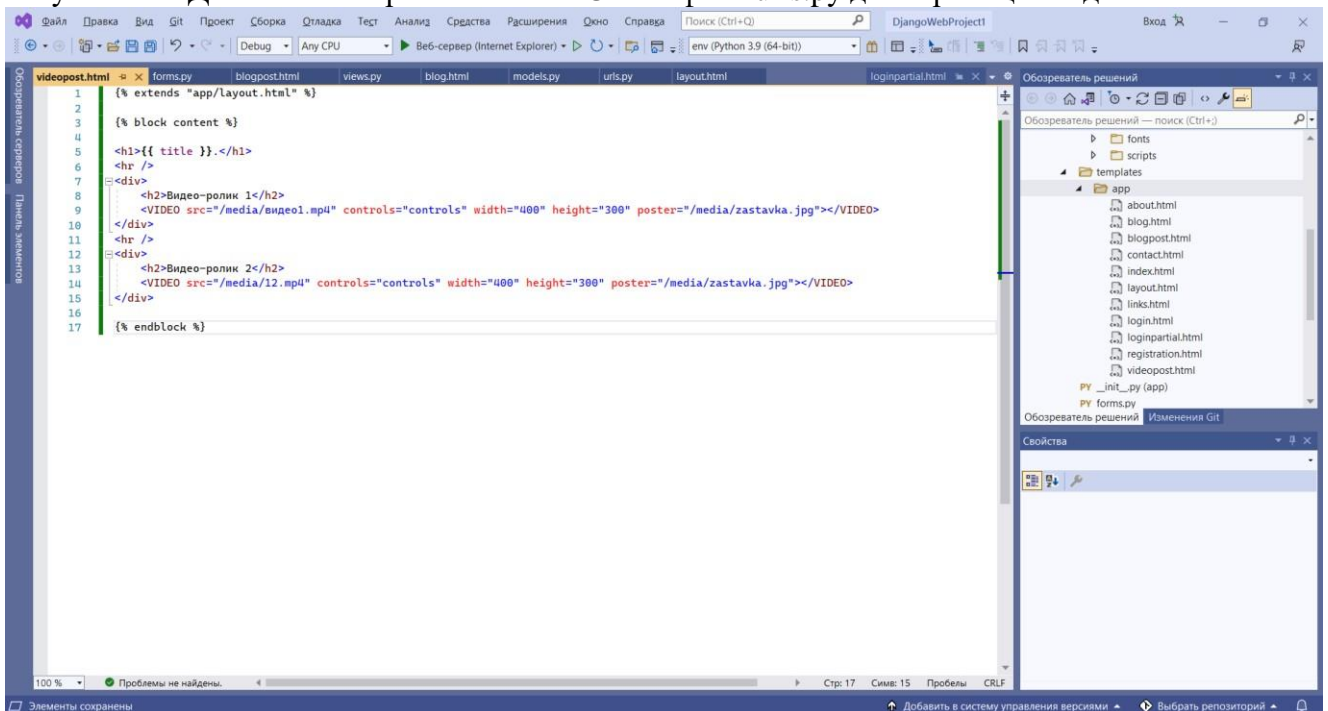


Рисунок 15 – Добавление кода для отображения и воспроизведения 2 видео-роликов в videopost.html

Тесты к экзамену

ТЕСТ 1 - ОДИН ПРАВИЛЬНЫЙ ВАРИАНТ ОТВЕТА

1. Что такое веб-сайт?

-) Отдельная веб-страница
-) Совокупность логически связанных между собой веб-страниц
-) Изображение, видео или аудио, доступные в Интернете

2. На каком языке написаны веб-страницы?

-) JavaScript
-) Python
-) HTML

3. Какой язык используется для форматирования веб-страниц?
 -) CSS
 -) HTML
 -) JavaScript
4. Для чего можно использовать JavaScript на веб-страницах?
 -) Для создания внешнего вида веб-страницы
 -) Для придания интерактивности веб-документу
 -) Для хранения медиа-ресурсов
5. Что такое таблица стилей CSS?
 -) Язык гипертекстовой разметки HTML
 -) Программа для просмотра веб-страниц
 -) Дополнительная возможность для создания внешнего вида веб-страниц
6. Что такое URL?
 -) Адрес документа в Интернете
 -) Название браузера
 -) Язык программирования
7. Какие протоколы используются для доступа к веб-страницам?
 -) FTP и Telnet
 -) HTTP и HTTPS
 -) SMTP и POP3
8. Кто может быть владельцем веб-сайта?
 -) Только физические лица
 -) Только юридические лица
 -) Как физические, так и юридические лица
9. Какие программы используют для просмотра веб-страниц?
 -) Браузеры
 -) Текстовые редакторы
 -) Графические редакторы
10. Что такое IP-адрес?
 -) Строка, которая указывает на местоположение документа
 -) Единообразный определитель местонахождения ресурса
 -) Название домена в Интернете
11. Какая среда разработки удобна для создания веб-сайтов на языке Python с использованием библиотеки Django?
 -) Visual Studio
 -) PyCharm
 -) Atom
 -) Sublime Text
12. Что является клиентским приложением при создании веб-сайтов на основе Веб 2.0?
 -) Браузер
 -) Веб-сервер
 -) Операционная система
 -) База данных
13. Где работают веб-серверы при создании веб-сайтов на основе Веб 2.0?
 -) На серверах в DATACenter
 -) На компьютерах пользователей
 -) На серверах в облаке
 -) На локальных серверах
14. Какие компоненты программного обеспечения должны быть установлены на веб-сервере для обеспечения функционирования веб-сайта?
 -) Операционная система
 -) Веб-сервер

-) База данных
 -) Интерпретатор скриптового языка
 -) Все варианты ответов верны
15. Какая задача у веб-клиента?
-) Хранение html-документов веб-сайта
 -) Генерация html-документов веб-сайта
 -) Загрузка и отображение html-документов и связанных с ним ресурсов
 -) Отдача html-документов клиенту
16. Какая задача у веб-сервера?
-) Хранение html-документов веб-сайта
 -) Генерация html-документов веб-сайта
 -) Загрузка и отображение html-документов и связанных с ним ресурсов
 -) Отдача html-документов клиенту
17. По какому протоколу происходит взаимодействие между веб-клиентом и веб-сервером?
-) DNS
 -) HTTPS
 -) FTP
 -) HTTP
18. Зачем нужен DNS-запрос при обращении к сайту?
-) Для получения IP-адреса, соответствующего доменному имени ресурса
 -) Для загрузки html-документов и связанных с ними ресурсов
 -) Для отправки запроса на сервер
 - d) Для отображения веб-страницы
19. Что происходит после получения искомого IP-адреса от DNS сервера?
-) Происходит отправка запроса на сервер
 -) Происходит загрузка html-документов и связанных с ними ресурсов
 -) Происходит отображение веб-страницы
 -) Происходит кэширование полученных данных
20. Как веб-сервер обрабатывает запрос от веб-клиента по протоколу HTTP?
-) Извлекает из запроса URL и отвечает на запрос клиента, возвращая html-документ
 -) Генерирует html-документ и отвечает на запрос клиента, возвращая его
 -) Хранит html-документы на сервере и отдает их клиенту по запросу
 -) Не обрабатывает запросы по протоколу HTTP
21. Что такое хостинг?
-) Услуга по предоставлению ресурсов для размещения информации на сервере
 -) Услуга по поиску сайтов в Интернете
 -) Услуга по созданию веб-сайтов.
22. Что такое Data center?
-) Аппаратное обеспечение сервера;
 -) Место, где находятся серверы хостинг-провайдера;
 -) Операционная система сервера.
23. Какой критерий выбора хостинга связан с программным обеспечением?
-) Размер дискового пространства;
 -) Количество месячного трафика;
 -) Используемая операционная система.
24. Что такое виртуальный выделенный сервер?
-) Сервер, на котором размещается только один сайт;
 -) Сервер, на котором размещается несколько сайтов, владельцы которых имеют одинаковые права и обязанности;

-) Автономная (выделенная) часть дискового пространства на сервере и фиксированные ресурсы, где владелец получает права администратора и может самостоятельно настраивать и устанавливать программы.
25. Что может влиять на быстродействие сервера при выборе хостинга?
-) Размер дискового пространства;
 -) Пропускная способность каналов;
 -) Количество месячного трафика.
26. Что может влиять на загрузку информации при выборе хостинга?
-) Свободные ресурсы процессора, оперативной памяти;
 -) Размер дискового пространства;
 -) Количество сайтов, которые можно разместить в рамках одной учётной записи.
27. Какие признаки могут использоваться для сравнения хостингов по качеству? -)
- Размер дискового пространства, количество месячного трафика;
-) Свободные ресурсы процессора, оперативной памяти, пропускная способность каналов;
 -) Используемая операционная система.
28. Что такое HTML?
-) Язык программирования для создания веб-сайтов
 -) Язык разметки гипертекста для описания содержания и оформления веб-страниц
 -) Язык для написания текстовых документов
29. В каком формате сохраняются HTML страницы?
-) .pdf
 -) .doc
 -) .html
30. Какие приложения отображают HTML документы?
-) Текстовые редакторы
 -) Браузеры
 -) Графические редакторы
31. Какой консорциум разрабатывает стандарты, имеющие отношение к Web?
-) ISO
 -) W3C
 -) IEEE
32. Какой последний по времени стандарт языка HTML?
-) HTML 4.01
 -) HTML 5.0
 -) HTML 5.2
33. Какие компоненты могут включаться в HTML документы?
-) Только текст
 -) Только картинки
 -) Картинки, таблицы, формы, мультимедиа данные, включая видео и звук
34. Что определяют теги в HTML документах?
-) Содержание документа
 -) Структуру и внешний вид документа при его отображении браузером
 -) Количество символов в документе
35. Что такое теги в HTML?
-) Это комментарии, которые не отображаются на странице
 -) Это элементы языка разметки, которые влияют на отображение страницы браузером
 -) Это специальные символы, которые используются для форматирования текста
36. Что такое контейнерные теги?
-) Теги, которые содержат только одну секцию
 -) Теги, которые содержат две связанные секции: открывающую и закрывающую, задающие область действия тега

-) Теги, которые используются для добавления картинок на страницу
- 37. Какой символ используется перед именем закрывающего тега?
 -) Обратный слеш
 -) Прямой слеш
 -) Двоеточие
- 38. Какой тег используется для добавления картинки на страницу?
 -)
 -)
 -) <i>
- 39. Что такое атрибуты тегов?
 -) Это имена тегов, которые используются для форматирования текста
 -) Это свойства, которые задают дополнительные параметры тегов
 -) Это комментарии, которые не отображаются на странице
- 40. Какие атрибуты используются для тега ?
 -) src и alt
 -) href и title
 -) class и id
- 41. Можно ли указывать имена тегов и атрибутов в любом регистре?
 -) Да
 -) Нет
- 42. Какие свойства имеют блочные теги?
 -) Они создают отдельные блоки, которые занимают всю ширину страницы
 -) Они размещаются внутри блоков
 -) Они являются одиночными тегами
- 43. Можно ли вкладывать контейнерные теги друг в друга?
 -) Да
 -) Нет
- 44. Можно ли вкладывать блочные теги в строчные теги?
 -) Да
 -) Нет
- 45. Какой тег используется для создания горизонтальной линии?
 -) <hr>
 -) <div>
 -) <p>
 -)

- 46. Какой тег используется для создания нумерованного списка?
 -)
 -)
 -)
 -) <p>
- 47. Какой тег используется для создания маркированного списка?
 -)
 -)
 -)
 -) <p>
- 48. Какой тег используется для создания заголовка?
 -) <p>
 -) <div>
 -) <h1>...<h6>
 -)
- 49. Какой тег используется для создания текстового абзаца?
 -) <h1>...<h6>

-) <p>
 -) <div>
 -)
50. Какой тег используется для создания блока предварительно отформатированного текста?
-) <pre>
 -) <code>
 -) <p>
 -)
51. Какой тег используется для создания гиперссылки?
-)
 -) <a>
 -)
 -)
52. Какой тег используется для выделения важного текста?
-)
 -) <code>
 -) <pre>
 -)
53. Какой тег используется для создания переноса строки?
-)

 -) <hr>
 -) <p>
 -) <div>
54. Какой тег задает тип документа?
-) <doctype>
 -) <html>
 -) <title>
 -) <head>
55. Какой тег содержит секцию заголовка документа?
-) <head>
 -) <body>
 -) <title>
 -) <html>
56. Какой тег указывает заголовок страницы?
-) <head>
 -) <body>
 -) <title>
 -) <html>
57. Какой тег задает базовый адрес для относительных ссылок внутри документа?
-) <base>
 -) <meta>
 -) <link>
 -) <script>
58. Какой тег используется для сопровождение документа метаданными и часто задает кодировку документа?
-) <base>
 -) <meta>
 -) <link>
 -) <script>
59. Какой тег указывает связи этого документа с внешними файлами?
-) <base>

-) <meta>
 -) <link>
 -) <script>
60. Какой тег предназначен для включения в состав документа стилевых таблиц?
-) <base>
 -) <meta>
 -) <link>
 -) <style>
61. Какой тег предназначен для включения в состав документа программных скриптов?
-) <base>
 -) <meta>
 -) <link>
 -) <script>
62. Какой тег задает текст страницы?
-) <head>
 -) <body>
 -) <title>
 -) <html>
63. Какой тег является тегом гиперссылки?
-) <a>
 -)
 -) <div>
 -)
64. Какой атрибут тега <a> указывает адрес документа по ссылке?
-) src
 -) href
 -) alt
 -) style
65. Какой тег используется для создания ссылки на файлы текущего сайта?
-)
 -)
 -)
 -)
66. Какой тег используется для создания ссылки на другой ресурс?
-)
 -)
 -)
 -)
67. Какой тег предназначен для отображения изображений на веб-страницах?
-)
 -) <picture>
 -) <image>
 -) <photo>
68. Какой атрибут тега задает альтернативный текст, заменяющий картинку, когда она недоступна?
-) title
 -) alt
 -) src
 -) href
69. Какой тег добавляет таблицу на веб-страницу?
-) <table>

-) <tr>
 -) <td>
 -) <th>
70. Какой тег создает интерактивную форму, которая может содержать поля ввода, кнопки и другие элементы?
-) <form>
 -) <input>
 -) <button>
 -) <label>
71. Какой атрибут тега <form> задает адрес серверного скрипта, которому будет пересылаться содержание формы для обработки?
-) method
 -) action
 -) enctype
 -) type
72. Как называется последовательность символов, которая заменяет некоторые символы в силу своей служебной роли?
-) escape-последовательность
 -) специальная последовательность
 -) HTML-последовательность
 -) служебная последовательность
73. Как начинаются и заканчиваются все escape-последовательности?
-) с символа «%» и точки с запятой
 -) с символа «#» и точки с запятой
 -) с символа «&» и точки с запятой
 -) с символа «\$» и точки с запятой
74. Какую escape-последовательность нужно использовать для отображения символа «<» на веб-странице?
-) <
 -) >
 -) &
 -) "
75. Зачем использовать таблицы стилей Cascading Style Sheets (CSS)?
-) Для управления содержимым веб-страницы
 -) Для управления дизайном веб-страницы
 -) Для управления сервером
76. Что содержит таблица стилей?
-) Набор правил (стилей), описывающих содержимое веб-страницы
 -) Набор правил (стилей), описывающих дизайн веб-страницы
 -) Набор правил (стилей), описывающих работу сервера
77. Как применяются стили к элементам веб-страницы?
-) Описываемые стилем параметры автоматически применяются ко всем элементам веб-страницы
 -) Каждый стиль должен быть явно привязан к соответствующему элементу веб-страницы
 -) Оба ответа верны
78. Как можно включить таблицу стилей в HTML-код веб-страницы?
-) Вставить таблицу стилей внутрь тега <head>
 -) Вставить таблицу стилей внутрь тега <body>
 -) Вставить таблицу стилей внутрь тега <footer>
79. Что означает спецификация CSS 3?

-) Единое правило для всех таблиц стилей
 -) Отдельные независимые модули, поддерживаемые современными браузерами
 -) Модульная структура для каждой таблицы стилей
80. Где обычно размещаются таблицы стилей?
-) внутри тега <body>
 -) в отдельных файлах с расширением css
 -) в теге <head>
 -) в теге <html>
81. Как подключить таблицу стилей, находящуюся в отдельном файле?
-) через тег <body>
 -) через директиву @import
 -) через тег <link>
 -) через тег <style>
82. Где можно указать встроенный стиль?
-) только внутри тега <head>
 -) только внутри тега <body>
 -) в любом месте HTML-кода
 -) только в отдельном файле css
83. Какой стиль имеет приоритет при противоречии между встроенными стилями и стилями из таблицы стилей?
-) внутренняя таблица стилей
 -) встроенный стиль
 -) внешняя таблица стилей
 -) не зависит от порядка указания стилей
84. Какой стиль имеет приоритет при противоречии между внешней таблицей стилей и внутренней таблицей стилей?
-) внешняя таблица стилей
 -) внутренняя таблица стилей
 -) не зависит от порядка указания стилей
 -) оба стиля будут применены
85. Какой стиль имеет приоритет при противоречии между несколькими стилевыми классами, привязанными к одному тегу?
-) стилевой класс указанный правее
 -) стилевой класс указанный левее
 -) не зависит от порядка указания стилевых классов
 -) оба стилевых класса будут применены
86. Как можно сделать атрибут стиля приоритетным?
-) указав его в теге <head>
 -) добавив восклицательный знак после значения атрибута стиля
 -) привязав его к стилевому классу
 -) нет правильного ответа
87. Какой тег является корнем дерева HTML-тегов документа?
-) <head>
 -) <body>
 -) <html>
 -) <title>
88. Для какого тега обычно задаются стилевые правила?
-) <head>
 -) <body>
 -) <html>
 -) <title>

89. Могут ли стилевые правила, наследуемые от тега `<body>`, задаваться для тега `<html>`?
-) да
 -) нет
90. Какой синтаксис используется для определения стиля в CSS?
-) `<селектор> [атрибут стиля]`
 -) `<селектор> {атрибут стиля: значение;}`
 -) `<селектор> (атрибут стиля: значение;)`
91. Что такое селектор в CSS?
-) Атрибут стиля, задающий, к каким элементам страницы будет применяться стиль
 -) Имя тега, к которому будет применяться стиль
 -) Блок из фигурных скобок, содержащий описание стилевых свойств
92. Что такое комбинированные стили в CSS?
-) Стили, которые применяются только к определенным тегам
 -) Стили, которые применяются только к определенным стилевым классам
 -) Стили, которые применяются к тегам, удовлетворяющим сразу нескольким условиям
93. Как задать сразу несколько одинаковых стилей в CSS?
-) Перечислить селекторы через запятую
 -) Соединить селекторы знаком `<>`
 -) Сестринский селектор задается знаком `<->`
94. Что означает дочерний селектор в CSS?
-) Стиль применяется к элементу, указанному слева, если он непосредственно, без промежуточных родительских тегов, вложен в элемент, указанный справа
 -) Стиль применяется к элементу, указанному справа, если он непосредственно следует за элементом, указанным слева, и находится на том же уровне вложенности
 -) Стиль применяется к элементу, который находится между элементами, указанными слева и справа
95. Какие свойства в CSS определяют параметры компоновки и позиционирования элементов на странице?
-) `margin, border, padding`
 -) `position, display, visibility, float, z-index, width, height, top, left`
 -) `background-color, color, font-size, font-family`
96. Какие символы не могут содержаться в идентификаторах переменных, методов и классов в Python?
-) `@, $, %`
 -) `#, /,`
 -) `(,), {`
97. Какой оператор используется для целочисленного деления в Python?
-) `%`
 -) `//`
 -) `/`
98. Какие типы данных поддерживает Python?
-) только целые и вещественные типы данных
 -) целые и вещественные типы данных, а также комплексный тип данных
 -) только строки и логический тип данных
 -) только списки и словари
99. Какой тип данных представляет собой неизменяемую упорядоченную последовательность данных?
-) список
 -) словарь
 -) множество

-) кортеж
100. Как задать список в Python?
-) в круглых скобках
 -) в фигурных скобках
 -) в квадратных скобках
 -) через запятую без скобок
101. Как задать словарь в Python?
-) в круглых скобках
 -) в фигурных скобках
 -) в квадратных скобках
 -) через запятую без скобок
102. С какого индекса начинается индексация элементов упорядоченных последовательностей в Python?
-) с 0
 -) с 1
 -) с -1
 -) с -2
103. Какие компоненты входят в приложение, созданное с помощью Django?
-) Модель, Представление, Контроллер
 -) HTML, CSS, JavaScript
 -) Файлы с расширением .py, .html, .css
 -) SQL команды, SQL таблицы, SQL запросы
104. Какое преимущество имеет подход MVC при разработке приложений?
-) Быстрое выполнение SQL запросов
 -) Легкость написания HTML кода
 -) Свобода объединения компонентов приложения и возможность их независимой модификации
 -) Улучшение безопасности веб-приложений
105. Какая функция у компонента models.py?
-) Определяет, какой метод контроллера будет вызван для URL
 -) Формирует внешний вид каждой веб-страницы - представление
 -) Описывает таблицы базы данных в виде классов Python - модели
 -) Содержит логику отображения веб-страниц
106. Какой компонент отвечает за логику отображения веб-страниц?
-) models.py
 -) views.py
 -) urls.py
 -) HTML файлы
107. Какое расширение имеют файлы, содержащие код представления в Django MVC?
-) .html
 -) .py
 -) .js
108. Что такое переменные шаблона в представлении Django MVC?
-) Методы, определенные в классах модели данных
 -) Данные, выводимые на веб-странице
 -) Особые команды для управления формированием кода веб-страницы
109. Какие действия выполняют теги шаблона в Django MVC?
-) Добавляют в конкретное место шаблона веб-страницы каких-либо значений (переменные шаблона)
 -) Управляют формированием кода веб-страницы
 -) Создают интерфейс приложения

110. Какой тег используется для создания цикла в Django MVC?

-) {% if % }
-) {% for % }
-) {% while % }

111. Где располагаются файлы с кодом представления в Django MVC по умолчанию?

-) \app\static\app
-) \app\templates\app
-) \app\views\app

112. В каком файле в веб-проекте Django находится код контроллера?

-) models.py
-) views.py
-) urls.py
-) settings.py

113. Что такое маршрут в файле urls.py в веб-проекте Django?

-) объект класса HttpRequest
-) объект класса HttpResponse
-) настраиваемый URL, связанный с методом действия контроллера
-) дополнительный параметр, передаваемый в метод действия контроллера

Ответ: с) настраиваемый URL, связанный с методом действия контроллера

114. Что происходит, когда пользователь запрашивает страницу в веб-проекте Django?

-) Django вызывает первый шаблон URL в файле urls.py
-) Django вызывает соответствующий метод действия контроллера, связанный с URL-адресом в файле urls.py
-) Django выполняет код метода действия контроллера и возвращает объект класса HttpResponse
-) Django останавливается на первом подходящем шаблоне URL в файле urls.py

115. Что происходит, если в веб-проекте Django не найден подходящий шаблон URL?

-) Django вызывает первый шаблон URL в файле urls.py
-) Django вызывает исключение 404
-) Django выполняет код метода действия контроллера и возвращает объект класса HttpResponse
-) Django останавливается на последнем шаблоне URL в файле urls.py

116. Какой файл хранит настройки проекта в виде набора переменных?

-) manage.py;
-) init.py;
-) settings.py;
-) urls.py.

117. В какой папке расположены скрипты, обновляющие базу данных в соответствии с изменениями в моделях?

-) templates;
-) static;
-) migrations;
-) models.

118. Какой модуль содержит код используемых в проекте форм?

-) models.py;
-) views.py;
-) forms.py;
-) admin.py.

119. Какой файл используется для редактирования макета страницы приложения?

-) settings.py;
-) urls.py;

-) layout.html;
-) views.py.

120. Как называется утилита, которая позволяет выполнять различные действия над проектом в Django?

-) manage.py;
-) init.py;
-) settings.py;
-) urls.py.

121. Каким синтаксисом используется включение шаблонов в другие шаблоны в Django?

-) `{% include <template_path> %}`
-) `{% inherit <template_path> %}`
-) `{% use <template_path> %}`

122. Каким синтаксисом определяется базовый шаблон в наследовании шаблонов в Django?

-) `{% base <template_path> %}`
-) `{% extend <template_path> %}`
-) `{% include <template_path> %}`

123. Каким тегом в базовом шаблоне отделяются блоки?

- a) `{% block <block_name> %} ... {% endblock %}`
- b) `{% start <block_name> %} ... {% stop <block_name> %}`
- c) `{% div <block_name> %} ... {% enddiv %}`

124. Какие области базового шаблона необходимо добавлять в дочерние шаблоны при наследовании в Django?

-) Блоки
-) Теги
-) Переменные

125. Что такое мастер-страница в веб-приложениях?

-) Шаблон веб-страницы, который определяет динамическое содержимое на всех страницах приложения
-) Шаблон веб-страницы, который определяет фиксированное содержимое для всех страниц приложения
-) Шаблон веб-страницы, который определяет статическое содержимое на всех страницах приложения

126. Что такое мастер-страница в веб-разработке?

-) Файл, который определяет динамическое содержимое на всех страницах приложения
-) Файл, который определяет фиксированное содержимое на всех страницах приложения
-) Файл, который определяет статическое содержимое на всех страницах приложения

127. Какой тег используется для обозначения места в мастер-странице, в которое будут подставляться другие представления?

-) `{% add %}`
-) `{% insert %}`
-) `{% block %}`

128. Какой код необходимо добавить в начало файла представления для подключения мастер-страницы?

-) { % add "app/layout.html" % }
-) { % use "app/layout.html" % }
-) { % extends "app/layout.html" % }

129. В какой папке по умолчанию находится мастер-страница в проекте, созданном в среде Visual Studio?

-) \app\templates\layout
-) \app\templates
-) \app\templates\app

130. Какая структура должна быть у файла мастер-страницы?

-) Содержать только теги <html>, <head>, <body>
-) Соответствовать обычному html-документу
-) Содержать только теги <head> и <body>

131. Какие теги используются для вставки других представлений в мастер-страницу?

-) { % add % }
-) { % include % }
-) { % insert % }

132. Какая директива используется для загрузки статичных файлов из папки static в мастер-страницу?

-) { % load staticfiles % }
-) { % include % }
-) { % block % }

133. Какой тег используется для подключения таблицы стилей в мастер-страницу?

-) <style>
-) <script>
-) <link>

134. Какой тег используется для создания кнопки раскрытия меню навигации при малых размерах окна браузера в мастер-странице?

-) <nav>
-) <button>
-) <a>

135. Какой класс используется для создания меню навигации по сайту в мастер-странице?

-) menu
-) nav
-) navigation

136. Как ограничить доступ пользователей к определенной странице в мастер-странице?

-) Использовать тег <restrict>
-) Использовать шаблонные теги { % if % }...{ % endif % } в представлениях
-) Использовать шаблонные теги { % block % }...{ % endblock % } в представлениях

137. Какой модуль Django нужно импортировать для создания веб-форм?

- a. django.models
- b. django.forms
- c. django.views
- d. django.templates

138. Как создать новую форму в Django?

- a. Создать новый класс формы, унаследованный от django.views.Form.
- b. Создать новый класс формы, унаследованный от django.models.ModelForm.
- c. Создать новый класс формы, унаследованный от django.templates.Form.
- d. Создать новый класс формы, унаследованный от django.forms.Form.

139. Какие поля можно использовать в форме Django?

- a. TextField, IntegerField, EmailField, DateField

- b. CharField, IntegerField, BooleanField, EmailField
 - c. StringField, DecimalField, FileField, DateField
 - d. FloatField, DateTimeField, TimeField, URLField
140. Как валидировать данные, введенные пользователем в форму Django?
- a. Нужно вручную проверять каждое поле формы.
 - b. Django автоматически валидирует данные.
 - c. Нужно использовать метод clean() в классе формы.
 - d. Нужно использовать функцию validate() в контроллере.
141. Как использовать JavaScript для валидации данных в форме Django?
- a. Нужно написать отдельную функцию для каждого поля формы.
 - b. Нужно добавить обработчики событий на каждое поле формы.
 - c. Нужно добавить обработчики событий на кнопку отправки формы.
 - d. Django не поддерживает использование JavaScript для валидации данных.
142. Что такое административный раздел Django?
- a. Система управления базами данных в Django.
 - b. Веб-интерфейс для управления контентом сайта в Django.
 - c. Библиотека для создания пользовательских интерфейсов в Django.
 - d. Механизм для обработки форм в Django.
143. Как войти в административный раздел Django?
- a. Нужно перейти по адресу /admin и ввести имя пользователя и пароль.
 - b. Нужно создать нового пользователя в административном разделе.
 - c. Нужно использовать командную строку для входа в административный раздел.
 - d. Django не поддерживает административный раздел.
144. Как настроить авторизацию в административном разделе Django?
- a. Нужно добавить новый класс авторизации в файл настроек проекта.
 - b. Нужно создать новую функцию для авторизации в файле views.py.
 - c. Нужно использовать стандартный механизм авторизации Django.
 - d. Django не поддерживает авторизацию в административном разделе.
145. Как настроить доступ к административному разделу Django для определенных пользователей?
- a. Нужно создать отдельную группу пользователей и назначить ее администратором.
 - b. Нужно использовать декоратор @login_required для контроллеров, отвечающих за доступ к административному разделу.
 - c. Нужно настроить соответствующие права доступа для каждого пользователя.
 - d. Django не позволяет настраивать доступ к административному разделу для определенных пользователей.
146. Какой модуль Django используется для работы с базами данных?
- a. django.models
 - b. django.forms
 - c. django.views
 - d. django.db

ТЕСТ 2 – НЕСКОЛЬКО ПРАВИЛЬНЫХ ВАРИАНТОВ ОТВЕТА

1. Какие программные средства могут использоваться для создания веб-сайтов?
-) Конструкторы сайтов
 -) CMS системы
 -) Средства разработки на языках программирования
 -) Ни один из перечисленных
2. Что выполняют браузеры при отображении текстов HTML документов?
-) Автоматический перенос строк текста в соответствии с шириной окна браузера
 -) Игнорируют символы перевода строки, вводимые с клавиатуры клавишей Enter

-) Отображают только один пробел, если слова в тексте разделяются несколькими пробелами
3. Что означает каскадирование в названии CSS?
-) Наследование стилей родительских элементов
 -) Применение стилей в порядке, имеющем разный приоритет
 -) Ни один из перечисленных
4. Какие селекторы могут выступать в качестве селекторов комбинированных стилей?
-) Только имена тегов
 -) Только имена стилевых классов
 -) Имена тегов, имена стилевых классов и имена стилей уникальных элементов
5. Какие структуры данных поддерживает Python?
-) только списки и словари
 -) кортежи
 -) списки
 -) словари
 -) множества
6. Какие операторы используются для управления последовательностью вычислений в программе?
-) Операторы цикла
 -) Операторы условия
 -) Операторы ввода/вывода
 -) Операторы математических вычислений
7. Какие компоненты входят в структуру MVC приложения?
-) Модель
 -) контроллер
 -) представление
 -) База данных
 -) сервер
 -) клиент
 -) HTML-код
 -) CSS
 -) JavaScript
8. Какие параметры принимает метод действия контроллера?
-) запрос и ответ
 -) запрос и список параметров
 -) объект класса HttpRequest и список параметров
 -) объект класса HttpRequest
 -) объект класса HttpResponse
9. Какие преимущества дает централизованное редактирование общих элементов на мастер-странице?
-) Ускорение процесса разработки
 -) Уменьшение вероятности ошибок при редактировании общих элементов
 -) Улучшение качества кода в целом
10. Какие элементы на мастер-странице могут быть динамическими?
-) Только футер
 -) Только содержимое страницы
 -) Содержимое страницы
 -) навигационная панель
11. Какие преимущества дает использование мастер-страницы в веб-разработке?
-) Уменьшение дублирования кода
 -) Централизованное редактирование общих элементов
 -) Большая гибкость в оформлении страниц

12. Какое свойство используется для создания отступов вокруг элемента?
- a) margin
 - b) padding
 - c) space
13. Какие структуры данных в Python являются изменяемыми?
-) tuple
 -) list
 -) set
 -) dict
14. Какие ключевые слова используются для определения класса и создания объекта в Python?
-) class
 -) object
 -) def
15. Какие типы данных в Python относятся к неизменяемым (immutable)?
-) list
 -) set
 -) tuple
 -) dict
16. Какие ключевые слова используются в Python для обработки исключений (ошибок)?
-) try
 -) except
 -) if
 -) else
 -) while
 -) for
17. Какие файлы содержатся в папке static в Django проекте?
-) Файлы с HTML-шаблонами
 -) Файлы с Python-скриптами
 -) Файлы с CSS-стилями
 -) JavaScript-скриптами
18. Какие элементы макета могут быть наследованы от мастер-страницы?
-) Цвета и фоны
 -) Шрифты и размеры текста
 -) Навигационное меню и футер
 -) Ни один из перечисленных
19. Какие преимущества предоставляет использование наследования шаблонов и мастер-страниц в веб-разработке?
-) Уменьшение количества кода
 -) Ускорение процесса разработки
 -) Улучшение совместимости и доступности сайта
 -) Ни один из перечисленных
20. Какие популярные CMS (системы управления контентом) используют наследование шаблонов и мастер-страницы?
-) WordPress
 -) Drupal
 -) Joomla
 -) Ни один из перечисленных
21. Как добавить пользователя в Django?

-) Используя консоль
 -) Используя административную панель
 -) Используя API
 -) Невозможно добавить пользователей в Django
22. Какой фреймворк Python часто используется для разработки Web 2.0 приложений?
-) Django
 -) Flask
 -) Pyramid
 -) Bottle
23. Какие из следующих принципов характерны для Web 2.0 приложений?
-) Разделение интерфейса и данных
 -) Кроссплатформенность
 -) Ориентация на пользователей
 -) Использование SOAP
 -) Ни один из перечисленных
24. Какая из следующих технологий используется для создания RESTful API в Python?
-) Flask-RESTful
 -) Django-RESTful
 -) Pyramid-RESTful
 -) Bottle-RESTful
25. Какой из следующих модулей Python используется для работы с базами данных?
-) SQLAlchemy
 -) Django ORM
 -) Peewee
 -) Flask-SQLAlchemy
26. Какая библиотека Python используется для создания графических интерфейсов пользователя?
-) PyQt
 -) Kivy
 -) Tkinter
 -) wxPython
27. Какой модуль Python используется для работы с датами и временем?
-) datetime
 -) time
 -) calendar
 -) dateutil
28. Какой метод Python используется для удаления последнего элемента из списка?
-) remove()
 -) pop()
 -) clear()
 -) del()
29. Какие из следующих языков программирования можно использовать для разработки веб-приложений?
-) Java
 -) C++
 -) Python
 - d-) Ruby
30. Какие из следующих баз данных используются для разработки веб-приложений?
-) MySQL
 -) Oracle

-) MongoDB
 -) SQLite
31. Какая команда используется для создания нового проекта Django?
-) `django-admin startproject myproject`
 -) `python manage.py startproject myproject`
 -) `startproject myproject`
 -) `python django-admin startproject myproject`
32. Какие из следующих методов Django используются для валидации данных формы?
-) `clean()`
 -) `validate()`
 -) `is_valid()`
 -) `check()`
33. Укажите два варианта действий, которые следует выполнить, чтобы добавить в веб-проект Django страницу сайта, содержащую форму?
-) Создать HTML-страницу с формой и добавить ее в проект.
 -) Создать класс формы, определить модель данных для формы и добавить представление для отображения формы.
 -) Создать функцию обработки запросов и добавить ее в `urls.py`.
34. Укажите два варианта действий, как передать переменную формы с данными в шаблон страницы?
-) Обращаться к полям формы через атрибут `request.POST` в шаблоне.
 -) Использовать объект контекста и передать в него переменную формы.
 -) Использовать функцию `render()` и передать в нее переменную формы.
35. Какие изменения необходимо внести, чтобы сделать русскоязычный интерфейс в веб-приложении?
-) Установить дополнительный пакет языковых файлов
 -) Изменить настройки языка в файле `settings.py`
 -) Создать файл перевода на русский язык
36. Какие расширения графических файлов могут быть у изображений, размещенных на веб-сайте?
-) `.jpg`
 -) `.png`
 -) `.gif`
 -) `.bmp`
37. Как преобразовать вводимые пользователем программы символы в числовые значения?
-) `int(input())`
 -) `str(input())`
 -) `float(input())`
 -) `eval(input())`
38. Как добавить на веб-страницу гиперссылки и графические изображения?
-) Использовать теги `<a>` и ``
 -) Написать ссылку или путь к изображению в коде страницы
 -) Вставить готовый HTML-код с гиперссылкой или изображением
39. Как задать цвет фона страницы?
-) Использовать свойство `background-color` в CSS-стилях
 -) Использовать атрибут `background-color` в теге `<body>` страницы
 -) Использовать атрибут `style` с `background-color` для каждого элемента на странице
40. Как добавить новую веб-страницу в проект?
-) Щелкнуть правой кнопкой мыши на проекте в окне "Обозреватель решений" и выбрать "Добавить" -> "Новый элемент"

-) Файл -> Создать -> Веб-страница
 -) Использовать комбинацию клавиш Ctrl - Shift - N
41. Опишите действия, которые необходимо выполнить для добавления в проект формы для ввода статьи блога.
-) Создать модель для статьи блога.
 -) Создать HTML-шаблон для отображения формы.
 -) Создать view для обработки данных из формы.
 -) Ни один из перечисленных
42. Из каких компонентов состоит онлайн-журнал?
-) Главная страница
 -) Рубрики и категории
 -) Статьи и материалы
 -) Каталог товаров
43. Какие функции может выполнять блог?
-) Предоставлять информацию о компании или личности
 -) Отображать новости и события
 -) Делиться опытом и знаниями
 -) Продавать товары и услуги
44. Опишите типичную структуру статьи блога.
-) Заголовок
 -) Вводный абзац
 -) Основная часть со списками, картинками или видео
 -) Заключительный абзац или вывод
 -) Комментарии и обратная связь
45. Какие существуют способы размещения списка статей блога на веб-странице?
-) Хронологический список
 -) Рубрикатор или категоризатор
 -) Список популярных статей
 -) Список статей с тегами
 -) Ни один из перечисленных
46. Какие инструменты могут использоваться для управления блогом?
-) CMS (системы управления контентом)
 -) Аналитика и статистика посещаемости
 -) Социальные сети и мессенджеры
 -) Инструменты для редактирования и обработки контента
47. Какова структура типичной модели Django?
-) Название модели
 -) Описание полей модели
 -) Методы модели
 -) Ни один из перечисленных
48. Для чего используются метаданные модели?
-) Для указания имени таблицы в базе данных
 -) Для указания порядка сортировки при выводе объектов модели
 -) Для указания прав доступа к модели
49. Каким образом в веб-проекте Django может быть реализована авторизация пользователей?
-) Использовать встроенные средства авторизации Django
 -) Создать собственную систему авторизации на основе базы данных
 -) Использовать сторонние библиотеки для авторизации
50. Какие базы данных используются на back-end?
-) PostgreSQL
 -) MySQL

-) Ruby

-) Swift

Тесты на установление правильной последовательности, в которых требуется указать правильный порядок действий или процессов.

1. Укажите правильный порядок действий при проектировании интерфейса пользователя:

a) Определение целей и задач пользователей

b) Создание макета интерфейса

c) Разработка иконок и графических элементов интерфейса

d) Определение требований к удобству использования и доступности интерфейса

2. Укажите правильный порядок действий при создании базы данных для веб-приложения:

a) Определение структуры базы данных

b) Создание таблиц и полей

c) Заполнение таблиц данными

d) Установка соединения с базой данных

3. Укажите правильный порядок действий при разработке сценария взаимодействия пользователя с интернет-приложением:

a) Определение требований к функционалу приложения

b) Создание схемы работы приложения

c) Определение последовательности действий пользователя

d) Разработка алгоритма работы приложения

4. Укажите правильный порядок действий при тестировании веб-приложения:

a) Подготовка тестовых данных

b) Запуск тестов на тестовом сервере

c) Анализ результатов тестирования

d) Исправление ошибок и повторное тестирование

5. Укажите правильный порядок действий при разработке документации к веб-приложению:

a) Определение требований к документации

b) Создание структуры документации

c) Написание текстовой части документации

d) Создание дизайна документации

6. Укажите правильный порядок действий при создании пользовательского интерфейса веб-приложения:

a) Разработка дизайна интерфейса

b) Создание макета интерфейса

c) Создание HTML-страниц интерфейса

d) Разработка функционала интерфейса

7. Установите правильную последовательность действий при проектировании базы данных:

-) Нормализация

-) Определение сущностей

-) Определение атрибутов

8. Установите правильную последовательность процессов при разработке интерфейса пользователя:

-) Создание прототипа

-) Окончательный дизайн

-) Проведение тестирования

9. Установите правильную последовательность действий при разработке web-страницы:

-) Добавление стилей CSS

-) Создание структуры HTML
 -) Добавление интерактивности JavaScript
10. Установите правильную последовательность действий при разработке серверной части web-приложения:
-) Определение API
 -) Разработка логики приложения
 -) Настройка сервера
11. Установите правильную последовательность процессов при тестировании web-приложения:
-) Написание тест-кейсов
 -) Отчетность
 -) Проведение тестирования
12. Установите правильную последовательность действий при разработке front-end части web-приложения:
-) Разработка интерфейса
 -) Создание дизайна
 -) Добавление интерактивности JavaScript
13. Какова структура веб-проекта Django?
-) models.py
 -) views.py
 -) urls.py
14. Какова правильная последовательность действий при использовании наследования шаблонов в Django?
-) Определение блоков в родительском шаблоне
 -) Определение блоков в дочернем шаблоне
 -) Определение наследования в дочернем шаблоне
15. Каков правильный порядок создания веб-страницы с помощью HTML?
-) Создание содержимого
 -) Добавление изображений
 -) Оформление стилей
16. Установите правильную последовательность типов данных в Python, от меньшего к большему:
-) bool
 -) str
 -) float
 -) list
 -) tuple
 -) int
 -) dict
17. Установите правильную последовательность действий при создании объекта класса в Python:
-) Определение класса
 -) Создание объекта
 -) Использование методов объекта
18. Какова правильная последовательность действий при использовании баз данных в Django?
-) Создание модели
 -) Создание шаблона
 -) Применение миграций
 -) Создание миграций
 -) Создание представления

19. Какая последовательность действий верна для создания веб-приложения на языке Python?

-) Создание файла HTML
-) Написание кода на Python
-) Создание файла Python
-) Запуск сервера

20. Какой порядок действий верен для подключения внешней CSS-стилевой таблицы на веб-странице?

-) Создание файла CSS
-) Написание кода на HTML
-) Подключение файла

21. Какая последовательность действий верна для добавления стилей на отдельный элемент на веб-странице?

-) Создание файла CSS
-) Написание стилей для выбранного элемента
-) Выбор элемента на веб-странице

22. Какой порядок действий верен для создания класса стилей в CSS?

-) Написание имени класса
-) Написание стилей для класса
-) Применение класса к элементу

23. Какая последовательность действий верна для создания гиперссылки на веб-странице?

-) Открыть тег <a>
-) Добавить атрибут href
-) Закрыть тег <a>
-) Добавить текст ссылки

24. Какой порядок действий является правильным для добавления изображения на веб-страницу?

-) Добавление тега
-) Добавление атрибута alt
-) Добавление атрибута src
-) Закрытие тега

25. Какой порядок действий верен для создания гиперссылки на изображение?

-) Добавление тега
-) Добавление тега <a>
-) Добавление атрибута href
-) Добавление атрибута src
-) Закрытие тега <a>
-) Закрытие тега

ТЕСТЫ НА УСТАНОВЛЕНИЕ СООТВЕТСТВИЯ

1. Сопоставьте каждую из действий с соответствующим ей пунктом процесса разработки интернет-приложения:

1. Анализ требований
2. Проектирование базы данных
3. Создание прототипа интерфейса пользователя
4. Тестирование
5. Разработка серверной логики

- A. Определение функциональности приложения
- B. Создание макетов страниц
- C. Создание моделей данных
- D. Проверка работоспособности приложения
- E. Написание кода на серверной стороне

2. Какой тип HTTP-запроса соответствует каждому из следующих действий?

- 1. Запрос на получение данных.
- 2. Запрос на создание новых данных.
- 3. Запрос на обновление существующих данных.
- 4. Запрос на удаление существующих данных.

- A. GET
- B. POST
- C. PUT
- D. DELETE

3. Свяжите каждый HTTP-статус с его описанием.

- 1. 200
- 2. 404
- 3. 500
- 4. 302

- A. ОК - запрос выполнен успешно
- B. Не найдено - запрошенный ресурс не найден
- C. Внутренняя ошибка сервера - сервер не смог обработать запрос
- D. Перенаправление - запрошенный ресурс перемещен в другое место

4. Свяжите каждый элемент URL-адреса с его описанием.

- 1. Протокол
- 2. Доменное имя
- 3. Путь
- 4. Параметры запроса

- A. Определяет, как будет происходить обмен данными.
- B. Имя узла, где находится веб-ресурс.
- C. Показывает, какую страницу или ресурс запрашивает клиент.
- D. Используется для передачи параметров веб-запроса.

5. Свяжите каждый тип формата данных с его описанием.

- 1. JSON
- 2. XML
- 3. YAML
- 4. CSV

- A. Формат хранения и передачи данных, основанный на текстовых файлах.

- В. Формат хранения и передачи данных, использующийся в различных приложениях.
- С. Формат хранения и передачи данных, использующий язык разметки.
- Д. Формат хранения и передачи данных, использующийся в API веб-приложений.

6. Соотнесите понятие с его определением:

- 1. CMS (система управления контентом)
- 2. FTP (протокол передачи файлов)
- 3. HTML (язык гипертекстовой разметки)
- 4. Ajax (асинхронный JavaScript и XML)
- 5. Хостинг (размещение сайта на сервере)
- А. Язык программирования, используемый для создания интерактивных веб-сайтов
- В. Интернет-протокол для передачи файлов между компьютерами
- С. Система, позволяющая управлять содержимым веб-сайта
- Д. Сервис, который предоставляет место для размещения веб-сайтов
- Е. Язык разметки, используемый для создания веб-страниц с помощью тегов
- Ф. Технология, которая позволяет обновлять содержимое веб-страницы без перезагрузки всей страницы

7. Установите соответствия между элементами двух множеств в HTML:

- 1. Тег для создания заголовка первого уровня
- 2. Тег для создания заголовка второго уровня
- 3. Тег для создания абзаца
- 4. Тег для создания списка
- 5. Тег для создания ссылки
- 6. Тег для создания изображения

- А. <h1>
- В. <h2>
- С. <p>
- Д.
- Е. <a>
- Ф.

8. Соотнесите элементы стиля CSS с их описанием:

- 1. Селектор
- 2. Свойство
- 3. Значение свойства

- А. Определяет, какой элемент HTML будет стилизован.
- В. Описывает, какие стили будут применены к выбранным элементам.
- С. Задаёт конкретные значения, которые будут использоваться для свойства.

9. Соотнесите элементы стиля CSS с их описанием:

- 1. Класс
- 2. Идентификатор

3. Стилевое правило

- A. Определяет стиль для выбранных элементов на основе класса, назначенного им в HTML-коде.
- B. Используется для определения стиля для конкретного элемента на основе значения атрибута "id" в HTML-коде.
- C. Определяет набор стилевых свойств для применения к выбранным элементам на основе селектора и значений свойств.

10. Соотнесите термины CSS с их описанием:

- 1. Селектор
- 2. Свойство
- 3. Значение
- 4. Блок объявлений

- A. Определяет элементы HTML, к которым будут применены стилевые правила.
- B. Определяет конкретный аспект визуального представления элементов HTML, который будет изменен.
- C. Определяет значение конкретного свойства CSS, которое будет применено к выбранным элементам HTML.
- D. Группа связанных между собой свойств, которые определяют визуальное представление элементов HTML.

11. Соедините элементы из первого множества с соответствующими элементами второго множества:

- 1. Синтаксис Python
- 2. Списки
- 3. Функции
- 4. Классы

- A. Позволяют создавать собственные типы данных и методы для работы с ними
- B. Используются для упорядоченного хранения и обработки коллекций элементов
- C. Определяются с помощью ключевого слова def
- D. Имеет динамическую типизацию и не требует явного объявления типа переменных

12. Соотнесите каждый элемент структуры веб-проекта Django с его описанием.

- 1. Модель (Model)
- 2. Представление (View)
- 3. Контроллер (Controller)
- 4. Файл маршрутов (urls.py)

- A. Хранит данные, используемые в приложении.
- B. Определяет, как данные будут представлены пользователю.

С. Обработывает запросы пользователя и обращается к нужным моделям и представлениям.

D. Определяет пути URL-адресов и их связь с представлениями.

13. Соотнесите следующие термины с их определениями:

1. Мастер-страница
2. Контентная страница
3. Шаблон
4. Наследование шаблонов

A. Основной документ, который содержит общую структуру и внешний вид всех контентных страниц.

B. Файл, содержащий HTML-код, который определяет внешний вид и расположение элементов на странице.

C. Механизм веб-разработки, который позволяет создавать новые шаблоны на основе существующих, сохраняя при этом все их особенности.

D. Страница сайта, которая использует шаблон для определения внешнего вида и расположения элементов.

14. Соотнесите следующие термины с их определениями:

1. Веб-форма a) Часть веб документа, совокупность элементов пользовательского интерфейса (полей ввода, флажков, списков и т.п.), позволяющих пользователю ввести и отправить на сервер определенные данные. Создается тегом form.
2. Поле формы b) Элемент HTML, обеспечивающий возможность пользовательского ввода или выбора. Поля ввода создаются тегами input, select и textarea.
3. Класс формы c) Класс на языке Python, производный от библиотечного класса forms.Form, состав полей которого является моделью содержания веб-формы.
4. Виджет d) Класс, задающий тег, который будет использован для создания поля формы в HTML-разметке.
5. Валидация e) Процедура проверки корректности пользовательского ввода данных в поля формы. Может выполняться на клиентской и на серверной стороне.

15. Соотнесите следующие термины с их определениями:

1. Аутентификация a) процесс определения личности пользователя
2. Авторизация b) процесс определения прав пользователя:
3. Приложение auth c) встроенная система аутентификации в Django, которая позволяет разработчикам добавлять аутентификацию и авторизацию в свои приложения
4. Объекты класса User d) представляют пользователей сайта и используются для проверки прав доступа, регистрации пользователей, ассоциации данных с пользователями

16. Соотнесите следующие термины с их определениями:

1. ORM a) отображение структур данных, описанных в программном коде, в таблицы реляционных баз данных и обратно
2. Модель b) класс на языке Python, с помощью методов которого можно создавать, получать, обновлять и удалять записи в таблице базы данных веб-приложения
3. Метаданные c) вложенный класс Meta, который задает дополнительные параметры модели

4. `objects`) экземпляр класса `QuerySet`, который представляет набор записей модели в базе данных
5. Поле модели `e)` поле, одного из объявленных в классе `Model` типа, представляющее столбец таблицы базы данных

17. Соотнесите следующие термины с их определениями:

1. Онлайн-журнал `a)` сайт для ведения онлайн-дневников (блогов), на котором можно публиковать свои и комментировать чужие записи, вести коллективные блоги
2. Блог `b)` сайт, который состоит из страницы со списком статей (постов) и страницы поста, которая включает полное содержание статьи, оставленные зарегистрированными пользователями комментарии и форму для добавления комментария.
3. Рубрика `c)` тематический раздел блога
4. Лента `d)` способ размещения контента на странице в одну колонку, в которой посты следуют друг за другом

18. Соотнесите каждый элемент формы в Django с его описанием:

1. `ModelForm`
2. `CharField`
3. `BooleanField`
4. `IntegerField`
5. `ChoiceField`

- A. Поле для ввода текста длиной до 255 символов
- B. Модель формы, автоматически созданная на основе модели Django
- C. Чекбокс для выбора да/нет
- D. Поле для ввода целого числа
- E. Выпадающее меню для выбора одного из нескольких вариантов ответа

19. Соотнесите каждый элемент формы в Django с его описанием:

1. `ModelForm`
2. `DateField`
3. `EmailField`
4. `DecimalField`
5. `HiddenInput`

- A. Поле для ввода email адреса
- B. Модель формы, автоматически созданная на основе модели Django
- C. Скрытое поле ввода
- D. Поле для ввода даты
- E. Поле для ввода десятичных чисел

20. Соотнесите каждую функцию Django ORM для работы с базами данных с ее описанием

1. `objects.create()`

2. `objects.filter()`
3. `objects.get()`
4. `objects.all()`

- A. Получение одного объекта из базы данных, возбуждение исключения в случае отсутствия объекта или нахождения более чем одного объекта.
- B. Создание нового объекта в базе данных и его сохранение.
- C. Получение всех объектов из базы данных.
- D. Получение объектов из базы данных, соответствующих заданному условию.

21. Соотнесите каждый тип поля в Django ORM с его описанием:

1. CharField
2. IntegerField
3. ForeignKey
4. DateTimeField

- A. Поле для хранения даты и времени.
- B. Поле для хранения целочисленных значений.
- C. Поле для хранения текстовых строк.
- D. Поле для связи моделей между собой.

22. Соотнесите каждый шаблонный тег Django с его описанием:

1. `{% if %}`
2. `{% for %}`
3. `{% url %}`
4. `{% block %}`

- A. Создает блок, который может быть переопределен в дочернем шаблоне.
- B. Позволяет создавать циклы для обхода коллекций данных.
- C. Создает ссылку на заданный URL-адрес.
- D. Позволяет создавать условия и ветвления в шаблонах.

23. Соотнесите элементы множеств:

- A. Django ORM
- B. Модель данных
- C. Атрибуты модели
- D. Поля модели
- E. Связи между моделями

1. Определяют структуру хранения данных.
2. Определяют связи между таблицами в базе данных
3. Позволяют управлять данными, используя SQL-подобный язык
4. Определяют типы данных, используемые в модели
5. Определяют способы взаимодействия между моделями

24. Соотнесите элементы веб-формы с их описанием:

1. <form>
2. <input>
3. <select>
4. <textarea>
5. <button>

- A. Элемент для создания выпадающего списка
- B. Элемент для создания кнопки отправки формы
- C. Элемент для создания текстового поля
- D. Элемент для создания многострочного текстового поля
- E. Элемент для создания формы

25. Соотнесите атрибуты формы с их описанием:

1. action
2. method
3. name
4. enctype

- A. Атрибут, указывающий метод отправки формы (GET или POST)
- B. Атрибут, определяющий URL-адрес, на который должна быть отправлена форма
- C. Атрибут, определяющий тип содержимого формы при отправке на сервер
- D. Атрибут, определяющий имя формы, которое используется для ссылки на нее и для отправки данных на сервер

ТЕСТЫ, ТРЕБУЮЩИЕ САМОСТОЯТЕЛЬНОГО ПОЛУЧЕНИЯ ОТВЕТОВ

1. Тег _____ используется для создания таблиц на веб-странице.
2. Для создания маркированного списка в HTML используется тег _____.
3. Тег _____ позволяет вставлять звуковые файлы на веб-страницу.
4. Атрибут _____ определяет адрес, по которому расположен файл с изображением.
5. Тег _____ используется для создания форм на веб-странице.
6. Атрибут _____ задает ширину рамки для элемента на веб-странице.
7. Тег _____ используется для создания ссылок на другие веб-страницы.
8. Атрибут _____ определяет значение, которое будет отправлено на сервер после отправки формы.
9. Использование _____ может помочь ускорить разработку интернет-приложений.
10. Для создания стилей на веб-странице используется язык _____.
11. _____ - это система контроля версий, которая позволяет отслеживать изменения в коде веб-приложения.
12. Размещение веб-сайта на сервере называется:
13. В Django структура проекта основана на шаблоне _____.
14. Шаблон MVC в Django реализуется как _____.
15. Использование мастер-страницы позволяет создавать _____ шаблон для всех страниц вашего сайта, что обеспечивает единообразный дизайн и расположение элементов на страницах.
16. Мастер-страница содержит общую информацию, такую как _____, которые могут использоваться на всех страницах, а также указания на _____, такие как местоположение основных меню и логотипа.

17. При создании новой страницы на основе мастер-страницы, вы можете внести изменения только в содержимое между _____, тогда как остальная часть страницы останется неизменной.
18. Django использует объектно-реляционную библиотеку для работы с базами данных, называемую _____.
19. _____ - способ размещения контента на странице в виде бэйджей в две или три колонки
20. _____ - класс, задающий тег, который будет использован для создания поля формы в HTML-разметке.
21. _____ - процедура проверки корректности пользовательского ввода данных в поля формы. Может выполняться на клиентской и на серверной стороне.
22. _____ - это высокоуровневый фреймворк для разработки веб-приложений на языке программирования Python.
23. В Django базу данных можно использовать с помощью модуля `django.db`, который предоставляет API для работы с различными _____, такими как PostgreSQL, MySQL и SQLite. Чтобы использовать базу данных в Django, необходимо создать модели, которые описывают структуру таблиц, а затем создать миграции и применить их к базе данных с помощью команды "migrate".
24. _____ файлы, такие как CSS, JavaScript и изображения, можно подключить в Django с помощью тегов `{% load static %}` и `{% static 'file_path' %}`, где `file_path` - это путь к файлу относительно папки "static" в проекте Django.
25. Для подключения статических файлов в Django нужно использовать тег `{% _____ %}` в шаблонах и определить путь к статическим файлам в файле настроек проекта (`settings.py`).