Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Владимирский государственный университет

имени Александра Григорьевича и Николая Григорьевича Столетовых»

(ВлГУ)

Колледж инновационных технологий и предпринимательства

**ОСНОВЫ РАБОТЫ С**

**ВЕБ-ФРЕЙМВОРКОМ DJANGO**

Методические указания

по лабораторным работам для студентов 3 курса

по дисциплине «Проектирование и дизайн информационных систем»

Составители:

Ларин Е.С.

Козлова Э.С.

Сорокина Е.М.

Владимир, 2023

**УДК 004.738.5 + 004.438 Python**

**ББК 32.973.26-018.1 Python я 73**

Авторы-составители:

Е.С. Ларин, преподаватель КИТП при ФГБОУВО «ВлГУ»;

Э.С. Козлова, редактор (балбесина 1)

Е.М. Сорокина, редактор (балбесина 2)

**Основы работы с веб-фреймворком Django:** Методические указания по лабораторным работам для студентов 3 курса по дисциплине «Проектирование и дизайн информационных систем» / сост. Е.С. Ларин, Э.С. Козлова, Е.М. Сорокина – Владимир: КИТП при ФГБОУВО «ВлГУ», 2023. - ?? с.

Методические указания составлены на основе учебной программы в рамках дисциплины «Проектирование и дизайн информационных систем» и представляют собой сборник теоретических основ и лабораторных работ, связанных с изучением и пониманием синтаксиса веб-фреймворка Django. Затрагивается разработка веб-сайтов на HTML и CSS, интеграция с Boodstrap и JavaScript, использование и настройка административного веб-сайта с использованием Django, работа с базами данных, создание CRUD операций и работа с npm-репозиторием. Подробно расписано создание полнофункционального веб-сайта.

В методических указаниях представлены и подробно описаны коды всех приведённых примеров, выполненных в среде разработки PyCharm Professional.

Предназначены для студентов 3 курса колледжа инновационных технологий и предпринимательства.

Ларин Е.С., 2023 г.

УДК 004.738.5 + 004.438 Python

ББК 32.973.26-018.1 Python я 73

КИТП, 2023

**Содержание**

**Лабораторная работы №1**

**Установка и настройка Django**

**Цель работы:** получить навыки первоначальной настройки и установки высокоуровневого Python веб-фреймворка Django

**Теоретические сведения:**

Фреймворк – это программная платформа, определяющая структуру программной системы, облегчающая разработку и объединение разных компонентов большого программного проекта. Простыми словами, это готовый набор инструментов, который помогает разработчику быстро создать продукт: сайт, приложение, интернет-магазин и т.д.

Django — это высокоуровневый веб-фреймворк, который позволяет быстро создавать безопасные и поддерживаемые веб-приложения с помощью языка программирования Python.

Django был разработан в период с 2003 по 2005 год командой, которая занималась созданием и обслуживанием газетных веб-сайтов. После создания нескольких сайтов, команда начала повторно использовать множество общего кода и шаблонов проектирования. Этот общий код эволюционировал в веб-фреймворк, который превратился в проект "Django" с открытым исходным кодом в июле 2005 года. На данный момент фреймворк продолжает развиваться.

Созданный опытными разработчиками, Django берёт на себя большую часть хлопот веб-разработки. Он бесплатный и с открытым исходным кодом, имеет растущее и активное сообщество, отличную документацию и множество вариантов как бесплатной, так и платной поддержки.

Принципы работы Django:

Django помогает писать программное обеспечение, которое будет: Полным – Django следует философии «Всё включено» и предоставляет почти всё, что разработчики могут захотеть сделать «из коробки». Поскольку всё, что вам нужно, является частью единого «продукта», всё это безупречно работает вместе, соответствует последовательным принципам проектирования и имеет обширную и актуальную документацию.

Разносторонним – Django может быть использован для создания практически любого типа веб-сайтов от систем управления контентом и wiki до социальных сетей и новостных сайтов. Он может работать с любой клиентской средой и может доставлять контент практически в любом формате (включая HTML, RSS-каналы, JSON, XML и т. д.).

Хотя Django предоставляет решения практически для любой функциональности, которая вам может понадобиться (например, для нескольких популярных баз данных, шаблонизаторов и т. д.), внутренне он также может быть расширен сторонними компонентами, если это необходимо.

Безопасным – Django помогает разработчикам избежать многих распространённых ошибок безопасности, предоставляя фреймворк, разработанный чтобы «делать правильные вещи» для автоматической защиты сайта. Например, Django предоставляет безопасный способ управления учётными записями пользователей и паролями, избегая распространённых ошибок, таких как размещение информации о сеансе в файлы cookie, где она уязвима или непосредственное хранение паролей вместо хэша пароля.

Django, по умолчанию, обеспечивает защиту от многих уязвимостей, включая SQL-инъекцию, межсайтовый скриптинг, подделку межсайтовых запросов и кликджекинг. Масштабируемым – Django использует компонентную “shared-nothing” архитектуру (каждая её часть независима от других и, следовательно, может быть заменена или изменена, если это необходимо). Чёткое разделение частей означает, что Django может масштабироваться при увеличении трафика, путём добавления оборудования на любом уровне: серверы кеширования, серверы баз данных или серверы приложений. Одни из самых загруженных сайтов успешно масштабировали Django (например, Instagram и Disqus).

Удобным в сопровождении – Код Django написан с использованием принципов и шаблонов проектирования, которые поощряют создание поддерживаемого и повторно используемого кода. В частности, в нём используется принцип «Don't Repeat Yourself» («не повторяйся»). Django также способствует группированию связанных функциональных возможностей в повторно используемые «приложения» и, на более низком уровне, группирует связанный код в модули (в соответствии с шаблоном Model View Controller (MVC))

PyCharm – это интегрированная среда разработки на языке Python, созданная компанией JetBrains. В ней есть все инструменты, чтобы писать, отлаживать и тестировать код.

Продукт доступен в двух версиях:

1. PyCharm Community Edition — бесплатная версия, находится под лицензией Apache License;
2. PyCharm Professional Edition — расширенная версия продукта, обладающая дополнительной функциональностью, является проприетарным ПО.

Далее будет подробно рассмотрено установка и настройка Django на обеих версиях PyCharm’а.

1. PyCharm Community Edition

Для начала необходимо скачать Python, если ещё не скачен, перейдя по ссылке:

<https://www.python.org/downloads/>

Вы можете установить любую версию, которую хотите, но лучшим вариантом будет установка одной из новейших версий. При установке Python’a необходимо установить следующий флажок. (см рис.1)

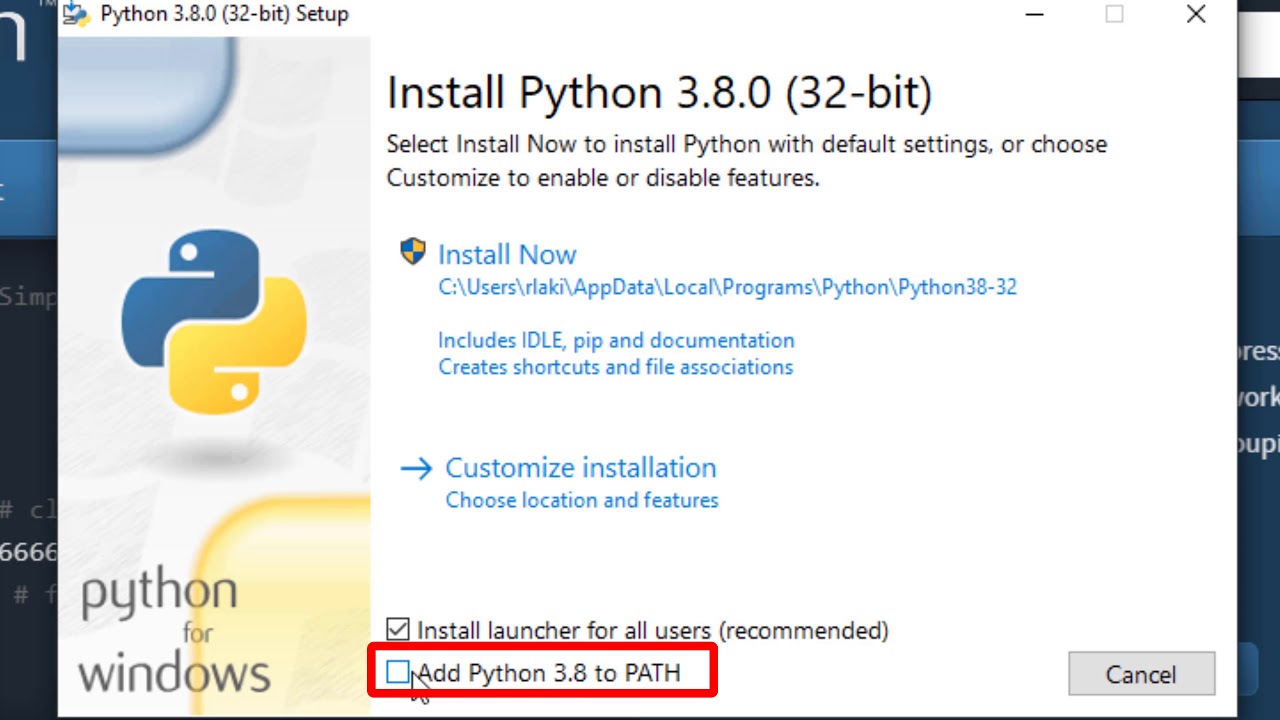


Рисунок 1. Установка Python

После установки PyCharma, открываем его и создаём новый стандартный проект с помощь. кнопки «New Project». (см рис.2)

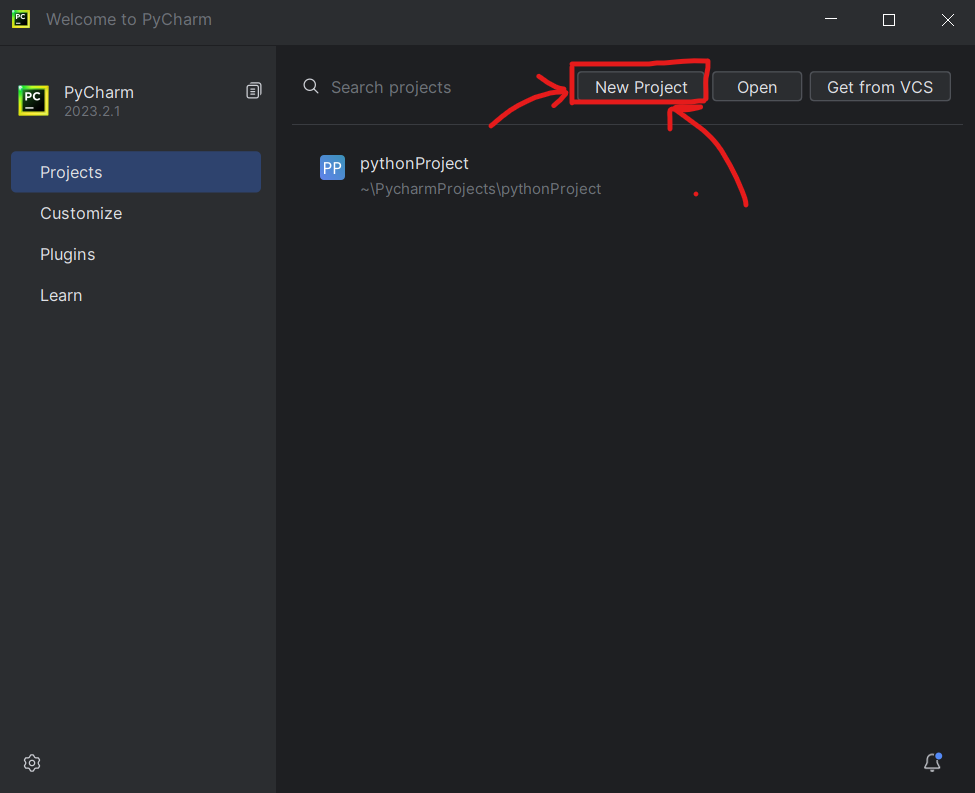


Рисунок 2.Создание нового проекта

Далее вы можете дать имя своему проекту и изменить его расположение, но лучше оставить его по умолчанию. После заполнения необходимо нажать на кнопку «Create», чтобы закончить создание проекта. (см рис.3-4)

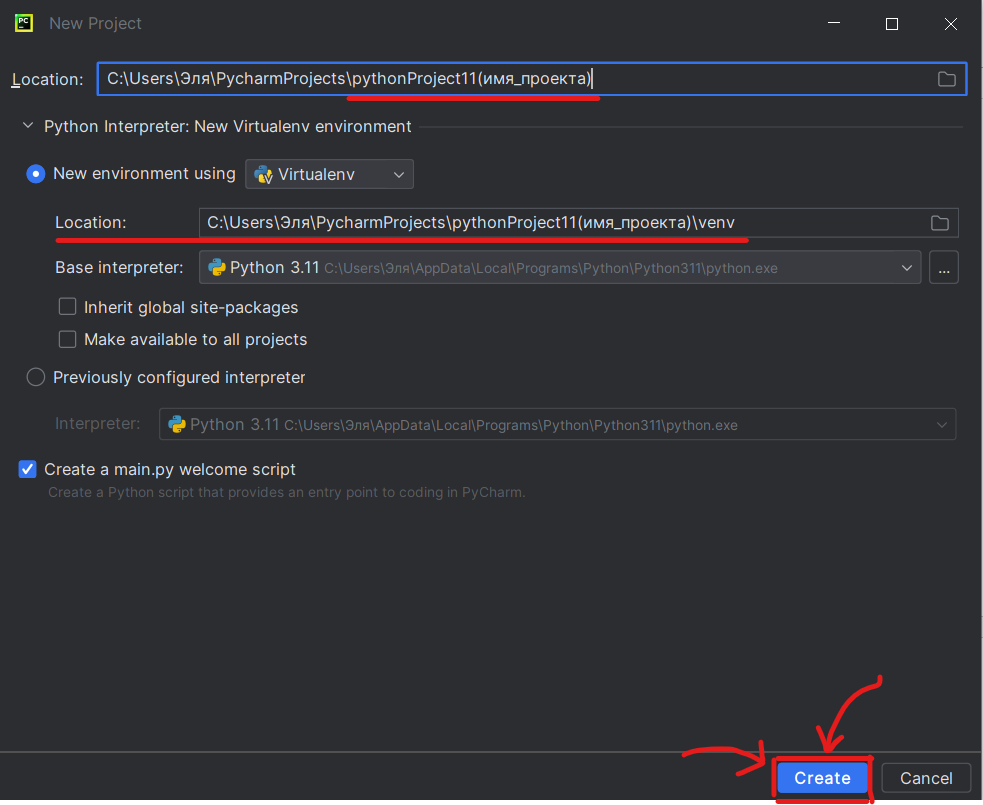


Рисунок 3.Свойства проекта

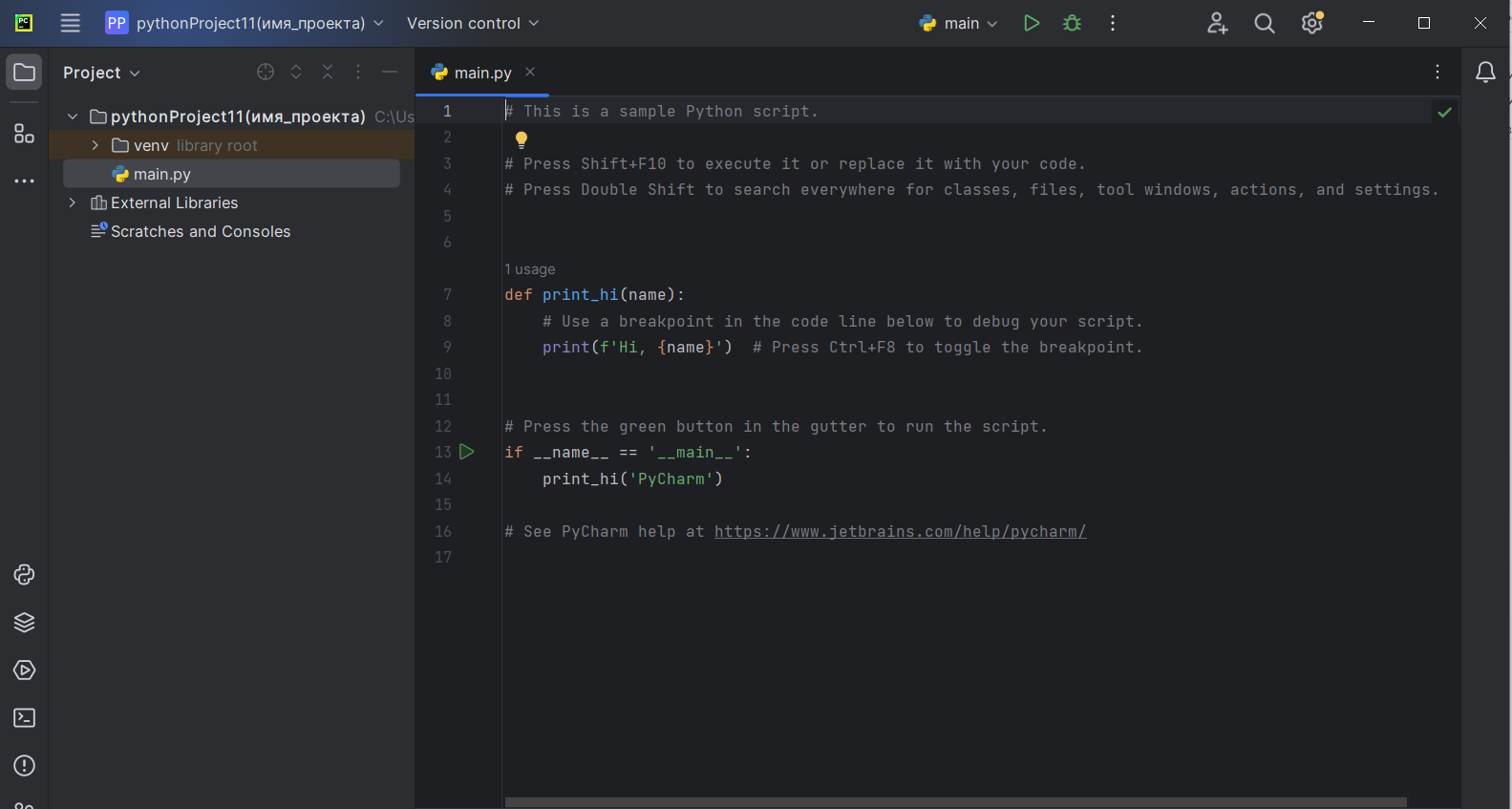


Рисунок 4.Созданный проект

После создания проекта устанавливаем ранее скаченный Python в PyCharm, для этого в нижней правой части окна находим название нашего проекта и yнажимаем на него правой копкой мыши, после чего появляется модульное окно, где выбираем «Add Interpreter», а затем «Add Local Interpreter». (см рис.5-6)

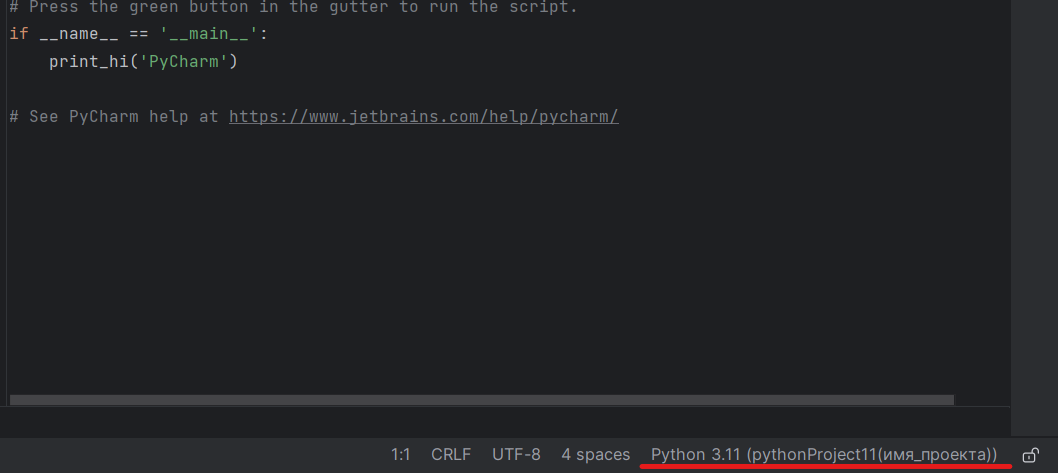


Рисунок 5.

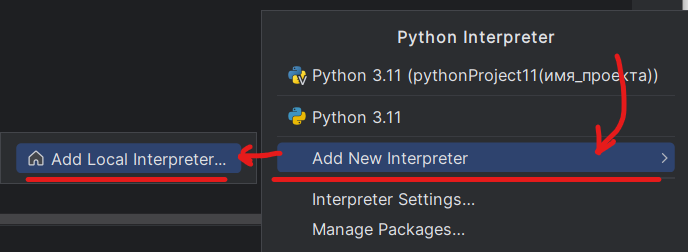


Рисунок 6.

Открывается страница добавления интерпретатора, где в меню мы переходим «System Interpriter» и в поле «Interpriter» выбираем установленный ранее Python и нажимаем кнопку «OK» (см рис.7-8)

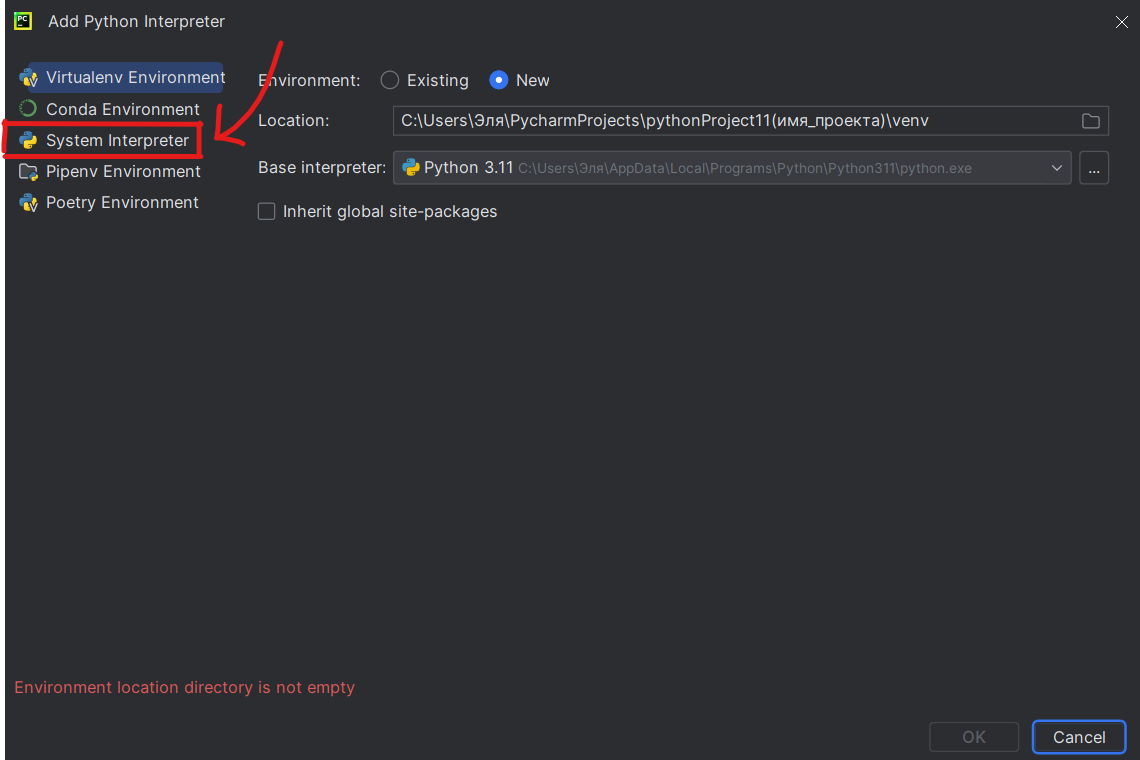


Рисунок 7

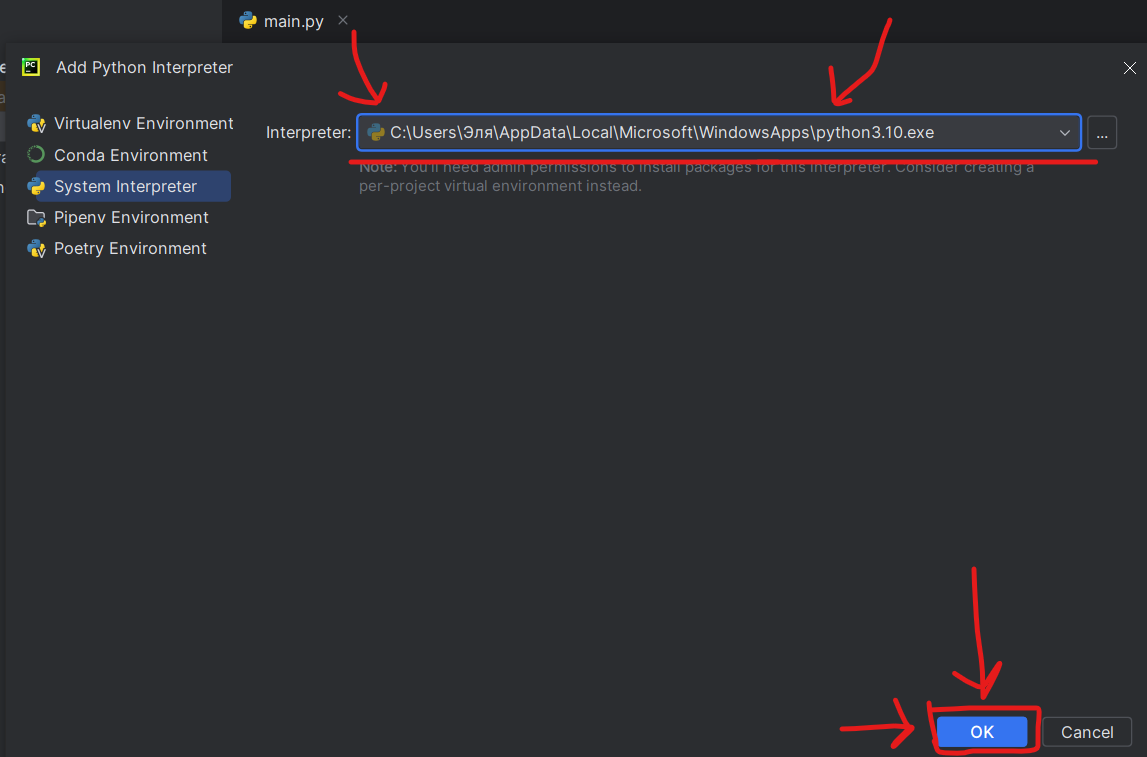


Рисунок 8. Подключение Python'a

Следующим шагом открываем терминал и скачиваем уже сам Django с помощью команды:

pip install Django

Открыть терминал можно в левой части окна в панели инструментов или с помощью комбинации клавиш Alt + F12 (см рис.9-10)

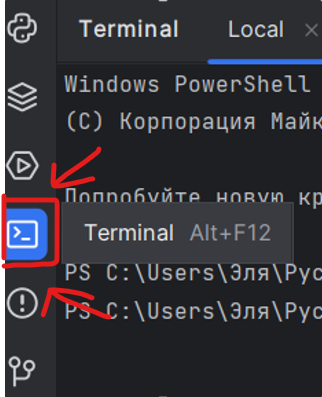


Рисунок 9. Кнопка терминала

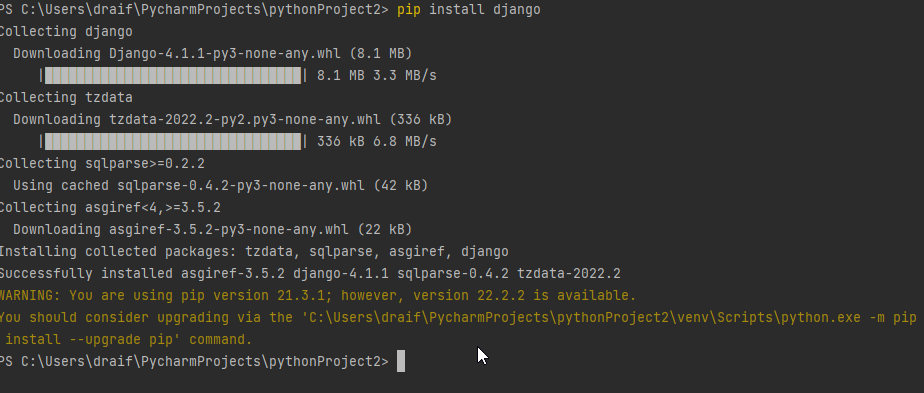


Рисунок 10. Установка Django

После установки необходимо создать само Django-приложение. Это делается с помощью ввода команды, которая вводиться в консоль:

django-admin startproject название\_вашего

В примере названием проекта является theatre. (см рис.11)



Рисунок 11. Создание django-приложения

**Должно произойти такое изменение, а именно появление вашего проекта в папке!** (см рис.12)

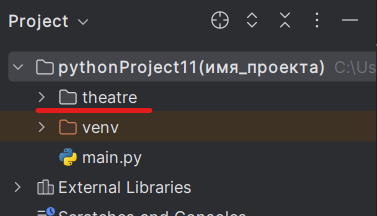


Рисунок 12. Обязательное изменение

Подключение к проекту осуществляется с помощью команды:

cd название\_вашего\_проекта

Для запуска проекта используется команда:

python manage.py runserver

После запуска проекта в консоль должна быть выведена информация о запуске проекта. (см рис.13)

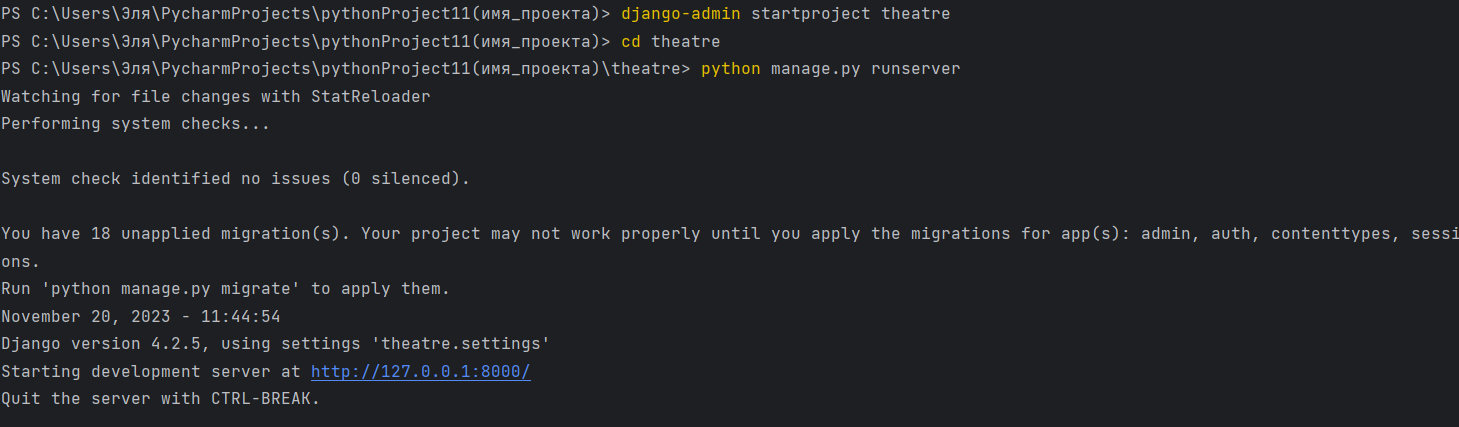


Рисунок 13. Запуск проекта

Последним шагом открываем проект в браузере по представленному адресу, должна появиться информационная страница представленная самим Django. (см рис.14-15)

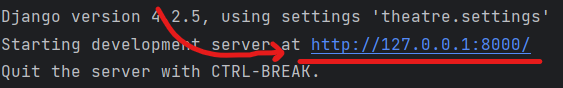


Рисунок 14. Доменное имя

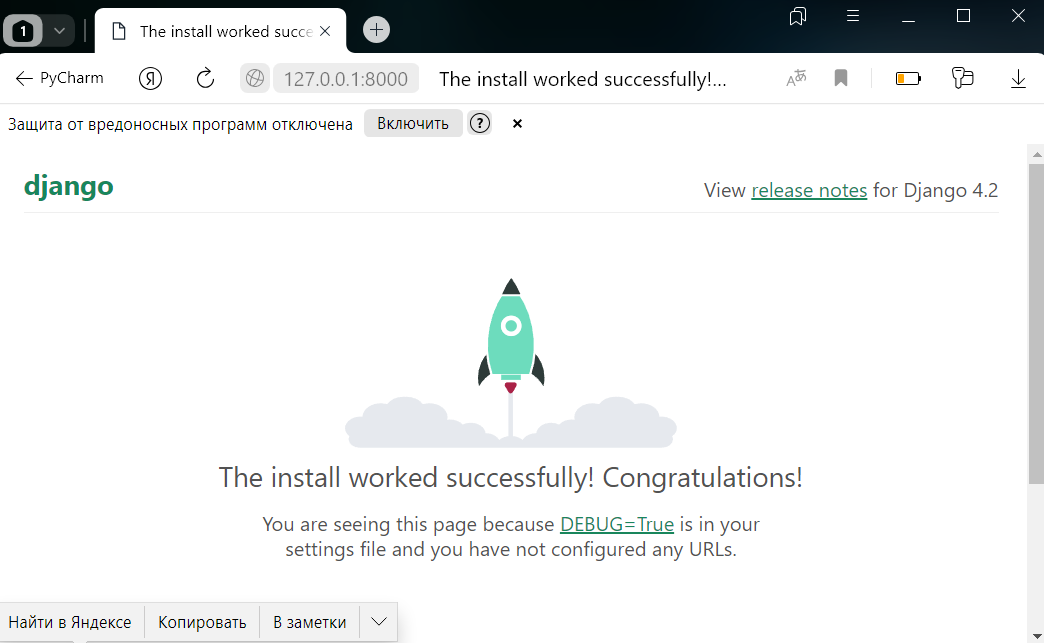


Рисунок 15. Вид страницы, которая должна открыться

2. PyCharm Professional

В профессиональной версии PyCharm’a создание Django-проекта немного упрощается.

Для начала открываем PyCharm и создаём новый проект, с помощью нажатия на «Create New Project»: (см рис. 16)

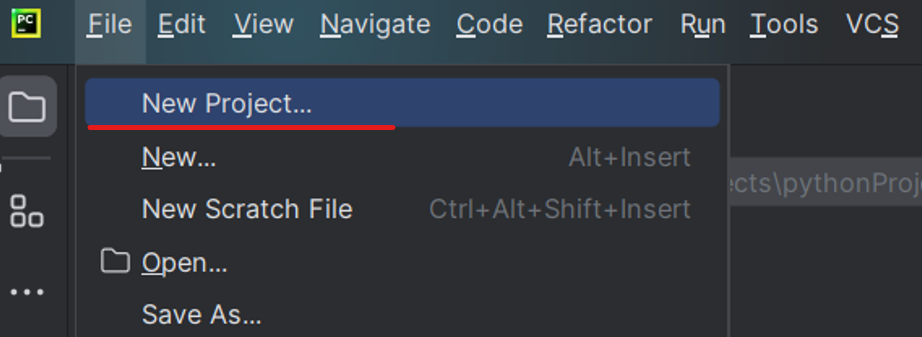


Рисунок 16.Создание нового проекта

Открылось окно создания нового проекта, где необходимо выбрать «Django» и выставить необходимые настройки проекта, т.е. его название и размещение, а затем нажать кнопку «Create». (см рис. 17)

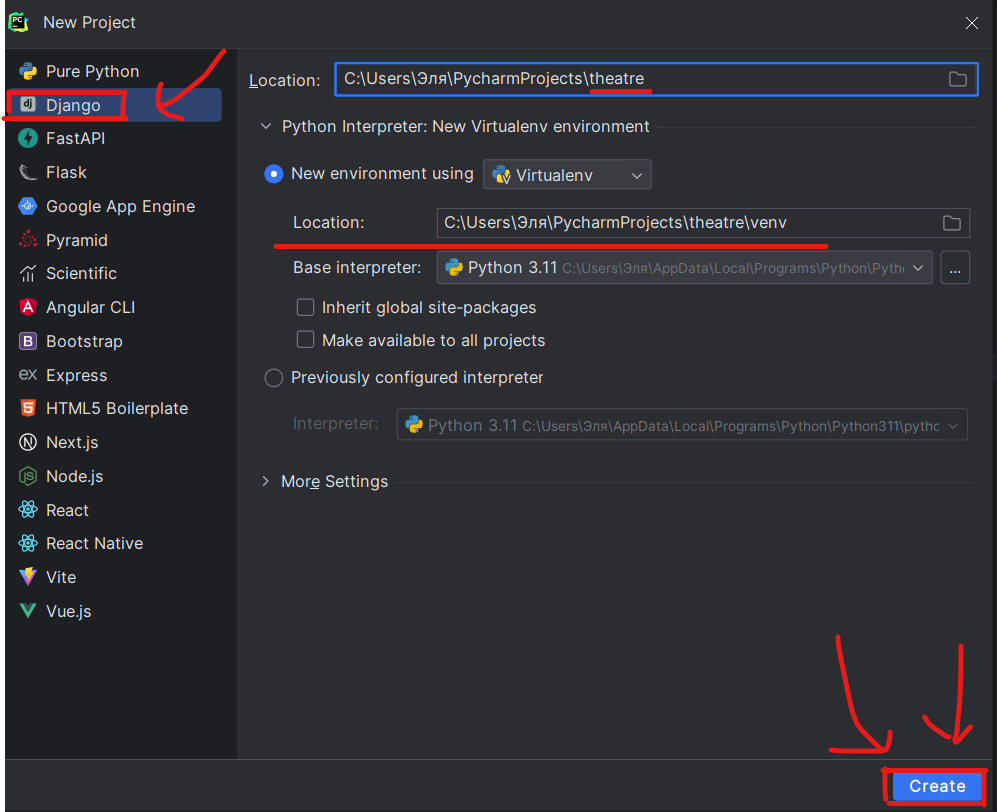


Рисунок 17.Настройка проекта

После нажатия кнопки начинается скачивание фреймворка и создание вашего проекта. (см рис. 18)

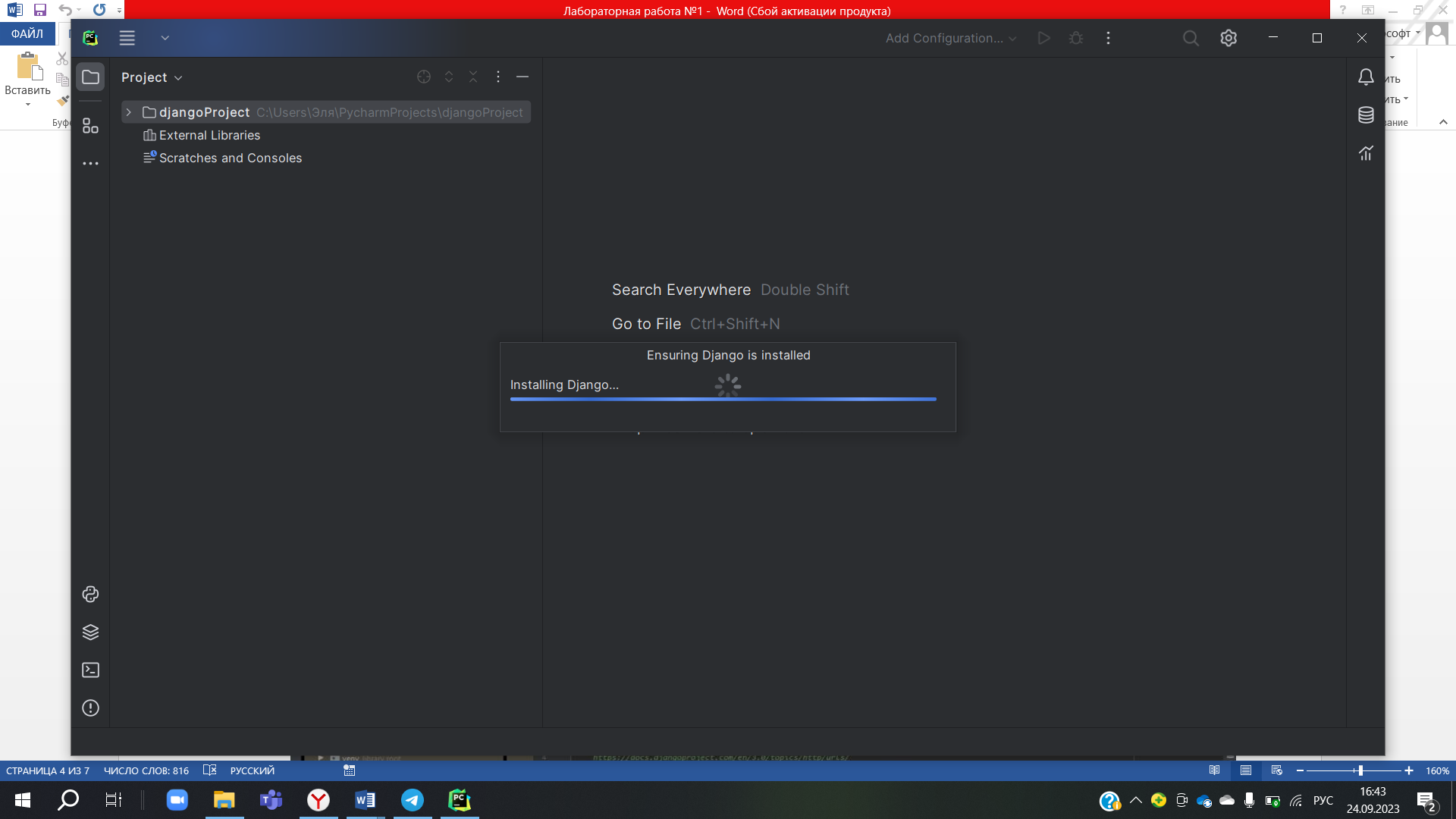


Рисунок 18.Скачивание фреймфорка и создание проекта

По окончанию загрузки открывается ваш созданный проект, который необходимо запустить с помощью «Run…». После нажатия на “Run…” выходит окно выбора для запуска проекта, где мы выбираем собственный проект. (см рис.19-20)

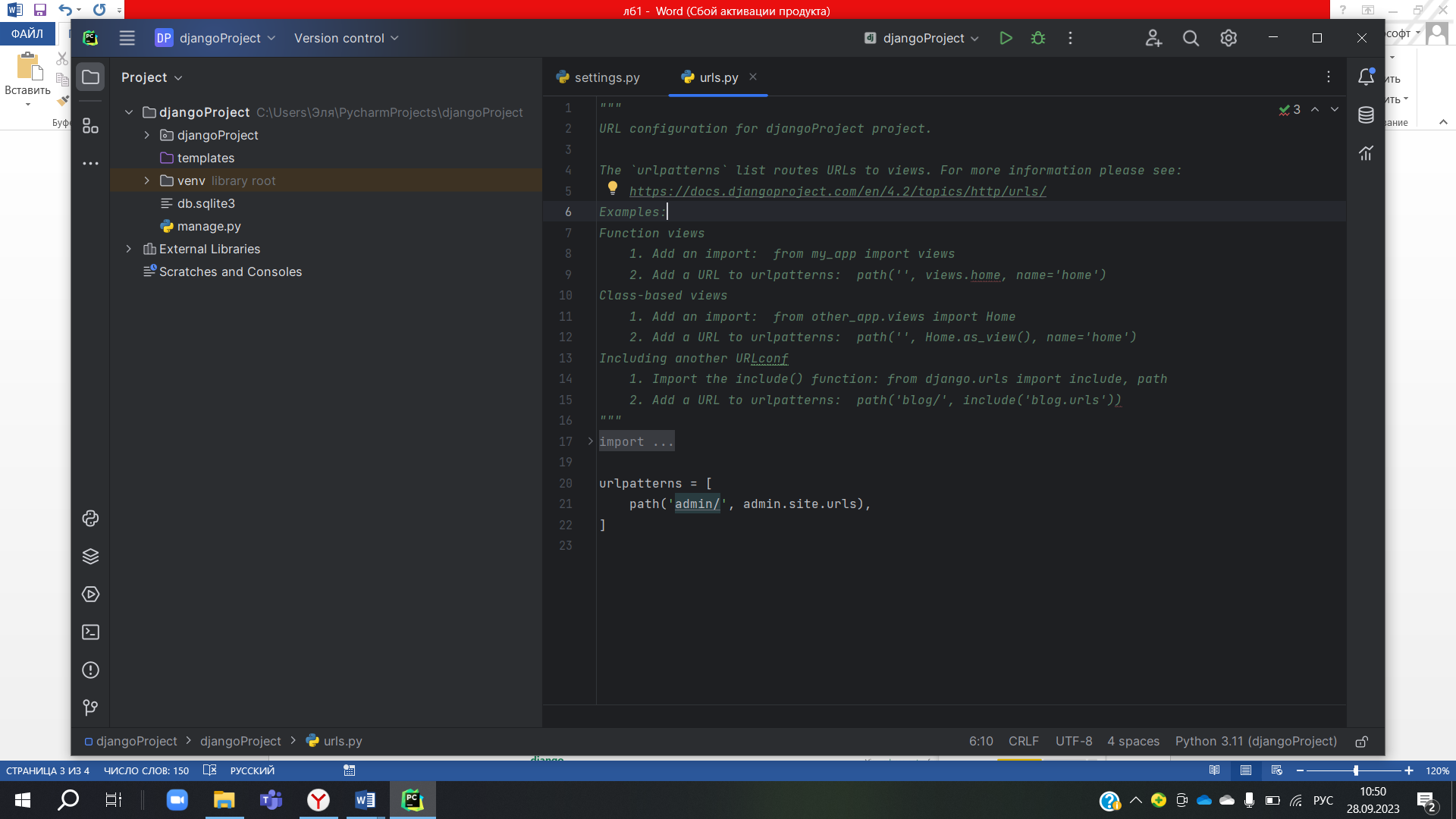


Рисунок 19.Созданный проект

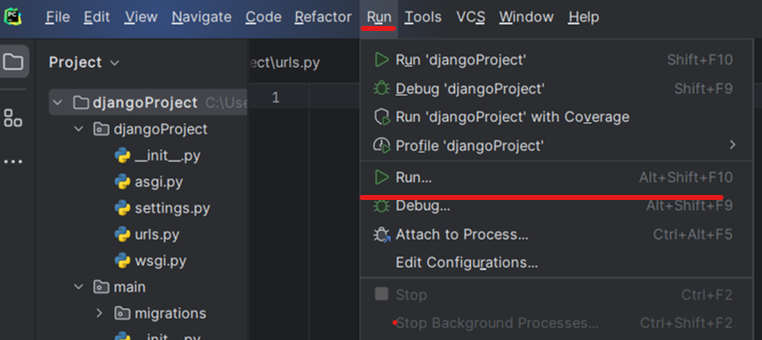


Рисунок 20.Запуск проекта

После запуска проекта в терминале выводятся данные запуска, в том числе адрес приложения, как в PyCharm Community Edition, поэтому шаги открытия проекта в браузере одинаковы для обеих версий.

**Задание для выполнения:**

1. Ознакомиться с теоретическими сведениями по фреймфорку Django;
2. Выбрать предметную область, для которой вы будете создавать свой Django-проект;
3. Создать виртуальную среду, проект и приложение по примеру, приведённому в лабораторной работе;
4. Оформить отчёт.

**Содержание отчёта:**

1. Титульный лист;
2. Цель работы;
3. Задание;
4. Ход работы (краткое описание предметной области, описание выполнения работы со скриншотами);
5. Вывод по работе.

**Контрольные вопросы:**

1. Что такое фреймворк?
2. Что такое Django?
3. Принципы работы Django?
4. Как создать Django проект?
5. Как запустить Django проект?

**Лабораторная работы №2**

**Настройка ссылок и создание страниц сайта**

**Цель работы:** научиться работать с ссылками в Django, а также способом вывода информации на страницы.

**Теоретические сведения:**

Маршрутизация – это

Path

Структура категории

Ход выполнения работы

**Задание для выполнения:**

1. Ознакомиться с теоретическими сведениями лабораторной работы;
2. По примеру создать собственную категорию в вашем проекте;
3. Оформить отчёт.

**Содержание отчёта:**

1. Титульный лист;
2. Цель работы;
3. Задание;
4. Ход работы (описание выполнения работы со скриншотами);
5. Вывод по работе.

**Контрольные вопросы:**

1. Что такое маршрутизация?
2. Как осуществляется маршрутизация в Django?
3. Как создать категорию в Django?
4. Из каких файлов состоит структура категории?
5. Каким способом осуществляется вывод информации на страницы сайта?