**Тема занятия № 4: Модуль 3. Создание и настройка проекта**

1. **Подготовка к работе**

Перед началом разработки сайта на Django, даже перед созданием его проекта, следует выполнять действия, перечисленные далее.

1. Проверить версии установленного ПО: исполняющей среды Python и серверных СУБД, если используемые проектом базы данных хранятся на них.

Установить сам фреймворк Django. Сделать это можно, указав в командной строке следующую команду:



Установить клиентские программы и дополнительные библиотеки для работы с СУБД.

Создать базу данных, в которой будут храниться данные сайта. Процедура ее создания зависит от формата базы.

Также Django поддерживает работу с базами данных Oracle, Microsoft SQL Server, Firebird, IBM DB2 и механизмом ODBC.

1. **Создание проекта Django**

Новый проект Django создается командой startproject утилиты django-admin, отдаваемой в следующем формате:

Django-admin startproject <имя проекта> [<путь к папке проекта>]

Если путь к папке проекта не указан, папка проекта будет создана в текущей папке и получит то же имя, что и сам проект. В противном случае папкой проекта станет папка с указанным в команде путем.

Папку проекта можно впоследствии переместить в любое другое место файловой системы, а также переименовать. Никакого влияния на работу проекта эти действия не оказывают.

1. **Настройки проекта**

Настройки проекта указываются в модуле settings.py пакета конфигурации. Значительная их часть имеет значения по умолчанию, оптимальные в большинстве случаев.

Настройки хранятся в обычных переменных Python. Имя переменной и есть имя соответствующего ей параметра.

1. **Основные настройки**

Основные настройки:

□ base dir — путь к папке проекта. По умолчанию вычисляется автоматически:

□ debug — режим работы сайта: отладочный (значение True) или эксплуатационный (False). По умолчанию — False (эксплуатационный режим), однако сразу при создании проекта для этого параметра указано значение True (т. Е. Сайт для облегчения разработки сразу же вводится в отладочный режим).

Если сайт работает в отладочном режиме, то при возникновении любой ошибки в коде сайта Django выводит веб-страницу с детальным описанием этой ошибки.

В эксплуатационном режиме в таких случаях выводятся стандартные сообщения веб-сервера наподобие ’’Страница не найдена” или ’’Внутренняя ошибка сервера”. Помимо того, в эксплуатационном режиме действуют более строгие настройки безопасности:

□ default charset — кодировка веб-страниц по умолчанию. По умолчанию: utf-8;

□ root urlconf — путь к модулю, в котором записаны маршруты уровня проекта, в виде строки. Значение этого параметра указывается сразу при создании проекта:

□ secret key — секретный ключ в виде строки с произвольным набором символов.

Используется программным ядром Django и подсистемой разграничения доступа для шифрования важных данных.

Значение параметра по умолчанию— пустая строка. Однако непосредственно при создании проекта ему присваивается секретный ключ, сгенерированный утилитой django-admin.

Менять этот секретный ключ без особой необходимости не стоит. Также его следует хранить в тайне, в противном случае он может попасть в руки злоумышленникам, которые используют его для атаки на сайт.

1. **Параметры баз данных**

Все базы данных, используемые проектом, записываются в параметре databases. Его значением должен быть словарь Python. Ключи элементов этого словаря задают псевдонимы баз данных, зарегистрированных в проекте. Можно указать произвольное число баз данных. Если при выполнении операций с моделями база данных не указана явно, то будет использоваться база с псевдонимом default.

В качестве значений элементов словаря также указываются словари, хранящие, собственно, параметры соответствующей базы данных. Каждый элемент вложенного словаря указывает отдельный параметр.

Значение параметра databases по умолчанию — пустой словарь. Однако при создании проекта ему дается следующее значение:

Оно указывает единственную базу данных, применяемую по умолчанию. База записывается в формате sqlite и хранится в файле db.sqlite3 в папке проекта.

Вот параметры баз данных, поддерживаемые Django:

□ engine — формат используемой базы данных.

Указывается как путь к модулю, реализующему работу с нужным форматом баз данных, в виде строки. Доступны следующие значения:

• django.db.backends.sqlite3— sqlite;

• dj ango.db.backends.mysql — mysql;

• dj ango.db.backends.postgresql —postgresql;

• dj ango. Db. Backends. Oracle — Oracle;

□ name — путь к файлу базы данных, если используется sqlite, или имя базы данных в случае серверных СУБД;

□ time zone — временная зона для значений даты и времени, хранящихся в базе.

Используется в том случае, если формат базы данных не поддерживает хранение значений даты и времени с указанием временной зоны. Значение по умолчанию — None (значение временной зоны берется из одноименного параметра проекта}.

Следующие параметры используются только в случае серверных СУБД:

□ host — интернет-адрес компьютера, на котором работает СУБД;

□ port — номер TCP-порта, через который выполняется подключение к СУБД. По

Умолчанию — пустая строка (используется порт по умолчанию);

□ user— имя пользователя, от имени которого Django подключается к базе данных;

□ password — пароль пользователя, от имени которого Django подключается к базе;

□ conn max age— время, в течение которого соединение с базой данных будет открыто, в виде целого числа в секундах.

Если задано значение 0, соединение будет закрываться сразу после обработки запроса. Если задано значение None, соединение будет открыто всегда. По умолчанию — 0;

□ options — дополнительные параметры подключения к базе данных, специфичные для используемой СУБД. Записываются в виде словаря, в котором каждый элемент указывает отдельный параметр. По умолчанию — пустой словарь.