**Краткое описание выбранных технологий:**

1. *Python.*

Python – это один из наиболее популярных современных языков программирования. Он пригоден для решения разнообразных задач и предлагает те же возможности, что и другие языки программирования: динамичность, поддержку ООП и кросс-платформенность.

Также Python является языком общего назначения, поэтому может применяться практически в любой области разработки ПО (standalone, клиент-сервер, Web-приложения) и в любой предметной области. Кроме того, Python легко интегрируется с уже существующими компонентами, что позволяет внедрять Python в уже написанные приложения.

1. *PyCharm*

PyCharm — интегрированная среда разработки для языка программирования Python. Предоставляет средства для анализа кода, графический отладчик, инструмент для запуска юнит-тестов и поддерживает веб-разработку на Django. PyCharm разработана компанией JetBrains[5] на основе IntelliJ IDEA.

PyCharm — это кросс-платформенная среда разработки, которая совместима с Windows, MacOS, Linux. PyCharm Community Edition (бесплатная версия) находится под лицензией Apache License, а PyCharm Professional Edition(платная версия) является проприетарным ПО.

1. *Django Framework*

Django — это про­грамм­ный пакет с различными воз­мож­но­стя­ми, под­хо­дя­щий для раз­ра­ботки слож­ных сай­тов и веб-при­ло­же­ний, напи­сан­ный на языке про­грам­ми­ро­ва­ния Python.

Django — [фрейм­ворк](https://web-creator.ru/articles/about_frameworks) для веб-при­ло­же­ний на языке [Python](https://web-creator.ru/articles/python). Один из основ­ных прин­ци­пов фрейм­ворка — [DRY](http://web-creator.ru/articles/dry) (don't repeat yourself). Веб-системы на Django стро­ятся из одного или несколь­ких при­ло­же­ний, кото­рые реко­мен­ду­ется делать отчуж­да­е­мыми и под­клю­ча­е­мы­ми. Это одно из замет­ных архи­тек­тур­ных отли­чий этого фрейм­ворка от неко­то­рых дру­гих (напри­мер, [Ruby on Rails](https://web-creator.ru/articles/about_ruby_on_rails)). Также, в отли­чие от мно­гих дру­гих фрейм­вор­ков, обра­бот­чики URL в Django кон­фи­гу­ри­ру­ются явно (при помощи регу­ляр­ных выра­же­ний), а не авто­ма­ти­че­ски зада­ются из струк­туры кон­трол­ле­ров.

Django про­ек­ти­ро­вался для работы под управ­ле­нием [Apache](http://web-creator.ru/articles/httpd) (с моду­лем mod\_python) и с использованием [PostgreSQL](https://web-creator.ru/articles/postgresql) в каче­стве базы дан­ных. В насто­я­щее вре­мя, помимо PostgreSQL, Django может рабо­тать с дру­гими СУБД: [MySQL](https://web-creator.ru/articles/mysql) ([MariaDB](https://web-creator.ru/articles/mariadb)), SQLite, Microsoft SQL Server, DB2, Firebird, SQL Anywhere и Oracle. Для работы [с базой дан­ных](https://web-creator.ru/articles/about_databases) Django исполь­зует соб­ствен­ный ORM, в кото­ром модель дан­ных опи­сы­ва­ется клас­сами Python, и по ней гене­ри­ру­ется схема базы дан­ных.

Архи­тек­тура Django похожа на [«Модель-Пред­став­ле­ние-Кон­трол­лер» (MVC)](https://web-creator.ru/articles/mvc). Кон­трол­лер клас­си­че­ской модели MVC при­мерно соот­вет­ствует уров­ню, кото­рый в Django назы­ва­ется Пред­став­ле­ние (View), а пре­зен­та­ци­он­ная логика Пред­став­ле­ния реа­ли­зу­ется в Django уров­нем Шаб­ло­нов (Templates). Из-за этого уров­не­вую архи­тек­туру Django часто назы­вают «Модель-Шаб­лон-Пред­став­ле­ние» (MTV).

Пер­во­на­чально раз­ра­ботка Django велась для обес­пе­че­ния более удобной работы с новост­ными ресур­са­ми, что доста­точно сильно отра­зи­лось на архи­тек­ту­ре: фрейм­ворк предо­став­ляет ряд сред­ств, кото­рые помо­гают в быст­рой раз­ра­ботке веб-сай­тов инфор­ма­ци­он­ного харак­те­ра. Напри­мер, раз­ра­бот­чику не тре­бу­ется созда­вать кон­трол­леры и стра­ницы для адми­ни­стра­тив­ной части сай­та, в Django есть встро­ен­ное при­ло­же­ние для управ­ле­ния содер­жи­мым, кото­рое можно вклю­чить в любой сайт, сде­лан­ный на Django, и кото­рое может управ­лять сразу несколь­кими сай­тами на одном сер­ве­ре. Адми­ни­стра­тив­ное при­ло­же­ние поз­во­ляет созда­вать, изме­нять и уда­лять любые объ­екты напол­не­ния сай­та, про­то­ко­ли­руя все совер­шён­ные дей­ствия, и предо­став­ляет интер­фейс для управ­ле­ния поль­зо­ва­те­лями и груп­пами (с пообъ­ект­ным назна­че­нием прав).

Веб-фрейм­ворк Django исполь­зуется в таких круп­ных и извест­ных сай­тах, как Instagram, Disqus, Mozilla, The Washington Times, Pinterest, lamoda и др.

Некото­рые воз­мож­но­сти Django:

* ORM, API доступа к БД с под­держ­кой тран­зак­ций
* встро­ен­ный интер­фейс адми­ни­стра­то­ра, с уже име­ю­щи­мися пере­во­дами на мно­гие языки
* дис­пет­чер URL на основе регу­ляр­ных выра­же­ний
* рас­ши­ря­е­мая система шаб­ло­нов с тегами и насле­до­ва­нием
* система кеши­ро­ва­ния
* интер­на­ци­о­на­ли­за­ция
* под­клю­ча­е­мая архи­тек­тура при­ло­же­ний, кото­рые можно уста­нав­ли­вать на любые Django-сайты
* «generic views» — шаб­лоны функ­ций кон­трол­ле­ров
* авто­ри­за­ция и аутен­ти­фи­ка­ция, под­клю­че­ние внеш­них моду­лей аутен­ти­фи­ка­ции: LDAP, OpenID и про­чие.
* система филь­тров («middleware») для постро­е­ния допол­ни­тель­ных обра­бот­чи­ков запро­сов, как напри­мер вклю­чён­ные в дис­три­бу­тив филь­тры для кеши­ро­ва­ния, сжа­тия, нор­ма­ли­за­ции URL и под­держки ано­ним­ных сес­сий
* биб­лио­тека для работы с фор­мами (насле­до­ва­ние, постро­е­ние форм по суще­ству­ю­щей модели БД)
* встро­ен­ная авто­ма­ти­че­ская доку­мен­та­ция по тегам шаб­ло­нов и моде­лям дан­ных, доступ­ная через адми­ни­стра­тив­ное при­ло­же­ние

Некото­рые ком­по­ненты фрейм­ворка между собой свя­заны сла­бо, поэтому их можно доста­точно про­сто заме­нять на ана­ло­гич­ные. Но с неко­то­рыми (напри­мер, с ORM) это сде­лать не очень про­сто. Кроме возможностей, встро­ен­ных в ядро фрейм­ворка, суще­ствуют паке­ты, рас­ши­ря­ю­щие его воз­мож­но­сти.

На базе Django раз­ра­бо­тано доста­точно много готовых реше­ний, рас­про­стра­ня­е­мых под свободной лицен­зи­ей, среди кото­рых системы для управ­ле­ния интернет-мага­зи­на­ми, универсальные [системы управ­ле­ния содержанием](https://web-creator.ru/articles/about_cms), а также более узко­на­прав­лен­ные проекты.