**Angular**

*Angular* - это *JavaScript* фреймворк имеющий большое комьюнити и являющийся одним из самых популярных фреймворков. *Angular* позволяет быстро создавать большие и сложные по части бизнес-логики приложения. Он обладает декларативным стилем кода, благодаря чему код становится более читаемый и его легче поддерживать. Также он обладает низким порогом входа и богатым набором функций. Существенно важным дополнением является то, что он обладает огромным количеством готовых решений, которые можно применять в своих проектах. *Angular* представляет не только инструменты, но и шаблоны дизайна для создания обслуживаемого проекта. При правильном создании *Angular* приложения у вас не будет путаницы классов и методов, которые сложно править и еще сложнее тестировать. Он позволяется веб-приложение под множество платформ. В заключении хочется сказать, что он является мощным инструментом, который подойдёт, как новичку, так и опытному программисту.

Минусами данного фреймворка является разнообразие различных структур *(Injectables, Components, Pipes, Modules* и т. д.) усложняет изучение по сравнению с *React* и *Vue.js*, у которых есть только «*Component*». Относительно медленная производительность, учитывая различные показатели. С другой стороны, это можно легко решить, используя так называемый «*ChangeDetectionStrategy*», который помогает вручную контролировать процесс рендеринга компонентов.

**Ruby on Rails**

*Ruby on Rails*  -  полноценный, многоуровневый фреймворк для построения веб-приложений, использующих базы данных, который основан на архитектуре Модель-Представление-Контроллер (*Model-View-Controller, MVC*). На *Ruby on Rails* написаны *GitHub, GitLab, AirBnB, Twitch, Shopify, Fiverr, Twitter*. Из отчественных — *InSales, UCHI.ru, Aviasales*. Что демонстрирует его популярность и важность в современных реалиях. Его основными преимуществами является:

1. **Скорость разработки**. Согласно статистике скорость разработки проектов на *RoR* увеличивается на 30–40 процентов по отношению к любому другому языку программирования или фреймворку. В первую очередь прирост скорости разработки определяется обширным набором готовых к работе штатных инструментов *RoR*, колоссальным набором готовых решений в сообществе, языку *Ruby* и простоте программирования на нем.
2. **Безопасность**. Ruby on Rails по умолчанию сильно заточены под безопасность проекта. При использовании инструментов RoR исключены SQL инъекции и XSS атаки. Все входные параметры экранируется по умолчанию. Выводимые переменные в шаблонах также экранируются, только если вы не указали обратной опции. У разработчика нет шансов допустить ошибки безопасности (не без исключений, разумеется).
3. **Стандарты.** Стандарты размещения файлов в проекте, стандарты написания кода в проекте, общие правила программирования в *Ruby on Rails* сильно структурируют любой проект. За счет этого проект становится читаемым. Вхождение в проект новичков происходит очень быстро. Опыт показывает, что любой новичок в проекте в первый же день работы делает свои первые полезные правки. За счет этого не считается большой проблемой, если разработку проекта изначально вела одна команда программистов, а поддержку проекта или доработку — совершенно другая. Проект на *RoR* априори понятен любому разработчику.

Его главными минусы это большое потребление ресурсов машины и сложность долговременной поддержки сайтов.

**Django**

*Django* считается лучшим веб-фреймворком, написанным на *Python*. В *Django* реализован принцип *DRY (don’t repeat yourself)*. Благодаря этому сокращается время создания сайтов. То есть при использовании *Django* вам не нужно несколько раз переписывать один и тот же код. Фреймворк позволяет создавать сайт из компонентов. *Django* был разработан, чтобы помочь разработчикам создать приложение настолько быстро, на сколько это возможно. Это включает в себя формирование идеи, разработку и выпуск проекта, где *Django* экономит время и ресурсы на каждом из этих этапов. Таким образом, его можно назвать идеальным решением для разработчиков, для которых вопрос дедлайна стоит в приоритете. Фреймвор может похвастаться внушительным списком дополнительных функций, позволяющих упростить процессы аутентификации пользователей и карт сайта, *RSS*, администрирования и так далее. Он работает с десятками дополнительных функций, которые заметно помогают с аутентификацией пользователя, картами сайта, администрированием содержимого, *RSS* и многим другим. Данные аспекты помогают осуществить каждый этап веб разработки. Работая в *Django*, вы получаете защиту от ошибок, связанных с безопасностью и ставящих под угрозу проект. Я имею ввиду такие распространенные ошибки, как инъекции *SQL*, кросс-сайт подлоги, *clickjacking* и кросс-сайтовый скриптинг. Для эффективного использования логинов и паролей, система пользовательской аутентификации является ключом. *Django* является большим и монолитным фреймворком. Это позволяет сообществу разрабатывать сотни универсальных модулей и приложений, но снижает скорость разработки самого *Django*. Кроме того, фреймворк должен поддерживать обратную совместимость, поэтому он развивается относительно медленно.

**Vue**

*Vue.js* — это известный термин в разработке современных веб-приложений. В настоящее время это одна из самых новых передовых технологий, которые часто упоминаются в связи с *Angular* и *React.js*. Существуют определенные преимущества использования Vue.js, которые должны побуждать разработчиков использовать его в своих проектах. Например, Vue.js во многих отношениях похож на *Angular* и *React*. *Vue.js* — отличный инструмент для разработчиков, потому что все его функции легко доступны. Для простоты использования разработчики могут легко назвать функцию так, как им нравится. Каждый сегмент может иметь отдельные функции, что облегчает настройку приложения в соответствии с индивидуальными требованиями. Одна из самых важных вещей — хорошая документация. Она показывает все варианты, предлагаемые платформой, и соответствующие примеры передового опыта. Документация на *Vue.js* постоянно совершенствуется и обновляется. Он также содержит простое вводное руководство и очень хороший обзор *API*. Возможно, это один из самых подробных документов, доступных для данного типа языка. *Vue* очень быстр. Возможно, он не самый быстрый, но его производительности с запасом хватает для подавляющего большинства веб-проектов. *Vue.js* помогает разрабатывать довольно крупные шаблоны для многократного использования, которые можно разработать без траты огромного количества времени в виду простой структуры. Существенным недостатком данного фреймворка является нехватка ресурсов. Vue.js по-прежнему имеет довольно небольшую долю рынка по сравнению с React или Angular. Это значит, что обмен знаниями в рамках фреймворка все еще формируется. Риск чрезмерной гибкости. Иногда у *Vue.js* могут возникать проблемы при интеграции в огромные проекты, а опыта о возможных решениях до сих пор нет. Но они обязательно появятся в ближайшее время.

**Bootstrap**

*Bootstrap* - один из самых популярных инструментов, который используется при создании сайтов и веб-приложений. *Bootstrap* включает в себя множество разных компонентов для веб-сайтов: типографику, веб-формы, кнопки, блоки навигации и другое.

Изначально *Bootstrap* был создан разработчиками Твиттера для внутреннего пользования, но спустя время он был выложен в открытый доступ и стал удобным набором инструментов для разработки пользовательских интерфейсов любой сложности. Он обладает высокой скоростью создания качественной адаптивной вёрстки даже начинающими веб-разработчиками (достигается это благодаря использованию готовых классов и компонентов, созданных профессионалами), наличие большого количество готовых хорошо продуманных компонентов, протестированных огромным сообществом веб-разработчиков на различных устройствах, низкий порог вхождения; для работы с фреймворком не обязательно иметь «глубокие» знания по *HTML*, *CSS*, *JavaScript* и *jQuery* (достаточно знать только основы этих технологий), наличие огромных сообществ и обучающих материалов; при желании это поможет не только хорошо разобраться в фреймворке, но и найти ответы практически на любые вопросы, кросс-браузерность и адаптивность. Сайты, созданные с помощью *Bootstrap*, идентично отображаются на разных устройствах и в современных браузерах.

К недостатком можно отнести шаблонность сайты, разработанные при помощи *Bootstrap*, похожи друг на друга: одинаковая структура, навигация, кнопки. Каждый новый сайт похож на множество уже созданных - и это не очень хорошо. Решить проблему можно - надо отказаться от использования готовых решений и максимально изменять шаблон в зависимости от пожеланий заказчика и идей дизайнеров. По этой причине важно разобраться во всех инструментах, предлагаемых *Bootstrap*, для того, чтобы максимально использовать их при разработке.

**Backbone**

*Backbone.js* – легковесный MVC-фреймворк, родившийся в 2010. Популярность набрал в качестве альтернативы тяжёлым фреймворкам вроде *ExtJS*. *Backbone* это *javascript*-библиотека для тяжёлых *frontend javascript* приложений, как например *gmail* или t*witter*. В таких приложениях вся логика интерфейса ложится на браузер, что в прочем даёт очень значительное преимущество в скорости интерфейса. В нём нет необходимости хранить данные в DOM, хранить данные в моделях, чрезвычайно полезная библиотека утилиты Underscore, код backbone.js хорошо документирован и отлично читается, Для Backbone нужен jQuery (либо его заменитель). К минусам можно отнести высокий порог входа.