



Министерство науки и высшего образования Российской Федерации  
Калужский филиал  
федерального государственного бюджетного  
образовательного учреждения высшего образования  
«Московский государственный технический университет имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(КФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИУК «Информатика и управление»

КАФЕДРА ИУК4 «Программное обеспечение ЭВМ, информационные технологии»

## НАУЧНО-ИССЛЕДОВАТЕЛЬСКАЯ РАБОТА «Исследование современных фреймворков»

Студент гр. ИУК4–72Б \_\_\_\_\_ ( Калашников А.С. )  
(подпись) (Ф.И.О.)

Руководитель \_\_\_\_\_ ( Гагарин Ю.Е. )  
(подпись) (Ф.И.О.)

Оценка руководителя \_\_\_\_\_ баллов \_\_\_\_\_  
30-50 (дата)

Оценка защиты \_\_\_\_\_ баллов \_\_\_\_\_  
30-50 (дата)

Оценка проекта \_\_\_\_\_ баллов \_\_\_\_\_  
(оценка по пятибалльной шкале)

Комиссия: \_\_\_\_\_ ( Гагарин Ю.Е. )  
(подпись) (Ф.И.О.)

\_\_\_\_\_ ( Белов Ю.С. )  
(подпись) (Ф.И.О.)

\_\_\_\_\_ ( Амеличева К.А. )  
(подпись) (Ф.И.О.)

Калуга, 2020

Калужский филиал федерального государственного бюджетного образовательного учреждения  
высшего образования

«Московский государственный технический университет имени Н.Э. Баумана  
(национальный исследовательский университет)»

(КФ МГТУ им. Н.Э. Баумана)

УТВЕРЖДАЮ

Заведующий кафедрой ИУК4  
(Гагарин Ю.Е.)

« 03 » сентября 2020 г.

## ЗАДАНИЕ на НАУЧНО-ИССЛЕДОВАТЕЛЬСКУЮ РАБОТУ (НИР)

За время выполнения НИР студенту необходимо:

1. Выполнить анализ информации по заданной теме, выявить наиболее рациональные варианты решения поставленной задачи; построить математическую модель исследуемого процесса, выполнить анализ объекта исследования посредством принятой математической модели; сформулировать выводы, адекватные полученным результатам.

*в том числе:*

*- провести анализ фреймворков Django, Spring MVC, Laravel и Ruby on Rails для разработки веб-приложения; рассмотреть их недостатки и преимущества; описать особенности фреймворков; показать различия исследуемых фреймворков; произвести выбор фреймворка в зависимости от требований к разрабатываемому проекту.*

2. Подготовить реферативный отчет о проделанной работе и защитить результаты НИР.

Дата выдачи задания « 03 » сентября 2020 г.

Руководитель НИР

\_\_\_\_\_

Гагарин Ю.Е.

Задание получил студент(ка) гр.ИУК4-72Б

\_\_\_\_\_

Калашников А.С.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
1. ЧТО ТАКОЕ ВЕБ-ФРЕЙМВОРК?.....	5
2. ВЕБ-ФРЕЙМВОРКИ.....	7
2.1. Django.....	7
2.2. Laravel.....	8
2.3. Spring MVC.....	10
2.4. Ruby on Rails.....	12
ЗАКЛЮЧЕНИЕ.....	15
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	16

## ВВЕДЕНИЕ

Веб-приложения беспрерывно завоевывают всё большую популярность. Их высокая гибкость, широкий выбор языков программирования и независимость от операционной системы клиента легко объясняют такой успех. Сама идея клиент-серверных приложений, будучи уже весьма зрелой, захватила мир программного обеспечения. Стандарты веб-разработки постоянно растут вместе со сложностью современных технологий. Сегодня актуальным является разработка веб-приложений, связанных с цифровой трансформацией предприятия. В результате цифровизации повышаются производительность каждого сотрудника и уровень удовлетворенности клиентов, и компания приобретает репутацию прогрессивной и современной организации.

Есть два пути разработки веб-приложений: с использованием фреймворков и без них. Некоторые веб-разработчики поддерживают идею отказа от фреймворков для разработки веб-приложений. В частности, речь идет о том, что веб-платформа настолько развита, что разработчику не нужны дополнительные API-интерфейсы для упрощения создания приложений. Разработка веб-приложений без использования фреймворков означает согласие не извлекать выгоду из работы групп разработчиков фреймворков, которые часто тесно сотрудничают с создателями браузеров. Другая часть разработчиков говорит о том, что фреймворки значительно упрощают и ускоряют разработку [1].

Цель работы - провести анализ фреймворков Django, Spring MVC, Laravel и Ruby on Rails для разработки веб-приложения.

Задачи:

1. Рассмотреть их недостатки и преимущества.
2. Описать особенности фреймворков.
3. Показать различия исследуемых фреймворков.
4. Произвести выбор фреймворка в зависимости от требований к разрабатываемому проекту.

## 1. ЧТО ТАКОЕ ВЕБ-ФРЕЙМВОРК?

Веб-фреймворки сильно изменили мир программирования и стали неотъемлемой частью процесса разработки.

Веб-фреймворк — инструмент, облегчающий процесс написания и запуска веб-приложения. Вам не нужно самостоятельно писать огромное количество строк кода и тратить время на поиск потенциальных просчётов и ошибок.

На рассвете эры веб-разработки все приложения писались вручную, и только разработчик приложения мог изменить или развернуть его. Веб-фреймворки позволили упростить жизнь разработчика. С 1995 года все работы, связанные с изменением структуры приложения, были приведены в порядок благодаря появлению общего подхода к разработке веб-приложений. В это время появились языки для веба. Сейчас их разнообразие позволяет выбрать подходящий как для статических, так и для динамических страниц. В зависимости от поставленной задачи, можно выбрать один фреймворк, покрывающий все нужды, или совместить несколько [2].

Слово «фреймворк» начали использовать примерно с начала XXI века. Если рассматривать перевод слова с английского - это «конструкция» или «структура».

Суть фреймворка заключается как раз в переводе слова. Это своеобразный каркас, используемый для того, чтобы существенно облегчить процесс объединения определенных компонентов при создании программ. Это основа, которая позволяет добавлять компоненты в зависимости от потребностей. База, на которой можно сформировать программу любого назначения достаточно быстро и без особых затруднений.

Если сравнивать динамическую библиотеку (DLL), которая отличается весьма ограниченным функционалом, и фреймворк, считающийся основой программ - можно выделить существенное преимущество фреймворков.

Именно фреймворк является связующим звеном, которое объединяет все

используемые программные компоненты. Также внутри фреймворка зачастую есть необходимые тематические библиотеки [3].

У фреймворков есть две основные функции: работа на серверной стороне (back-end) и работа на клиентской стороне (front-end).

Бэкенд-фреймворки отвечают за внутреннее устройство приложения. Правила и архитектура таких фреймворков не даёт возможности создать веб-приложение с богатым интерфейсом. Они ограничены в своей функциональности, однако вы всё равно можете создавать простые страницы и разные формы. Также они могут формировать выходные данные и отвечать за безопасность в случае атак. Всё это определённо может упростить процесс разработки. Серверные фреймворки в основном отвечают за отдельные, но критически важные части приложения, без которых оно не сможет нормально функционировать.

Фронтенд-фреймворки связаны с внешней частью приложения. Они отвечают за внешний вид приложения. В отличие от серверных, клиентские фреймворки никак не связаны с логикой приложения. Этот тип фреймворков работает в браузере. С их помощью можно улучшить и внедрить новые пользовательские интерфейсы. Фронтенд-фреймворки позволяют создавать разные анимации и одностраничные приложения. Все клиентские фреймворки отличаются по функциональности и использованию [4].

## 2. ВЕБ-ФРЕЙМВОРКИ

### 2.1. Django

Язык программирования Python считается одним из самых популярных и востребованных для создания веб-приложений. Он способен решать различные задачи, в том числе реализовывать веб-приложения, которые используют миллионы пользователей. «Дзен Python» - это собственная философия языка, на основе которой работает Python.

Django - это веб-фреймворк Python, использующий шаблон проектирования MVC.(рис.1) Django был выпущен в 2005 году и постепенно стал одним из лучших фреймворков для веб-разработки. Возможности фреймворка позволяют сразу приступить к разработке [5].

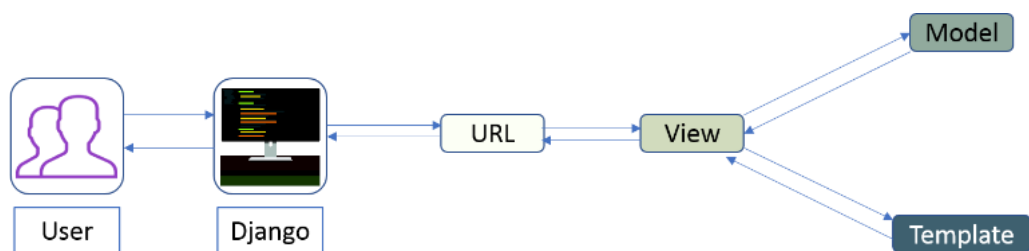


Рис. 1 Шаблон MVC в Django

Django обладает рядом преимуществ. Django был разработан, чтобы помочь разработчикам создавать приложения как можно быстрее. Этот подход предполагает создание идеи, разработку и выпуск проекта, где Django экономит время и ресурсы на каждом из этих этапов. Таким образом, данный фреймворк можно назвать идеальным решением для разработчиков, для которых вопрос дедлайна стоит на первом месте. Django работает с огромным количеством дополнительных функций, которые значительно упрощают реализацию аутентификации пользователей, помогают с картами сайта, администрированием контента, RSS и др. Во время работы вы получаете

защиту от ошибок, связанных с безопасностью, таких, как SQL-инъекции, clickjacking и т. д., которые ставят под угрозу проект. Фреймворк Django лучше всего подходит для работы с самыми высокими уровнями трафика.

Недостатками Django являются использование шаблона маршрутизации с URL, Django является слишком монолитным, все основано на ORM Django, компоненты разворачиваются вместе, необходимо освоить всю систему для работы.

Для ускорения рабочего процесса, можно установить индивидуальную конфигурацию. Раньше разработка кода требовала большого количества времени, теперь появилась возможность значительно упростить этот процесс[6].

## 2.2. Laravel

Laravel – это бесплатный PHP-фреймворк с открытым исходным кодом, предназначенный для разработки веб-приложений, использующий архитектурную модель MVC; он был выпущен в 2011 году, но благодаря быстрым темпам развития и огромному количеству поклонников сейчас это один из самых популярных PHP-движков.

К числу преимуществ Laravel можно отнести документацию. Документация Laravel очень структурирована. В ней Laravel есть отдельные статьи для каждой конструкции и каждого процесса. Поскольку у этой PHP- платформы много последователей по всему миру, вы можете найти множество различных сообществ, форумов и пользовательских переводов статей в сети. Структура кода соответствует популярному шаблону проектирования MVC.(рис.2) MVC позволяет сделать код более читаемым, процесс разработки становится комфортнее, дифференцируя работу frontend- и backend- разработчиков. Artisan - это консоль Laravel, которая позволяет использовать команды для работы с миграциями, контроллерами, моделями, авторизацией и другими базовыми компонентами фреймворка. Blade - это настраиваемый



шаблонизатор с набором собственных директив. После установки фреймворка Laravel в распоряжении разработчика есть файлы `app.js` и `app.css`, которые представляют собой скомпонованные и минимизированные jQuery и Bootstrap самых последних версий на момент выхода релиза Laravel. Laravel предоставляет механизм для регистрации и авторизации пользователей. В Laravel есть конструкции, позволяющие проводить валидацию данных на основе различных готовых правил. В Laravel можно создавать собственные правила, сообщения об ошибках и настраиваемые валидаторы. Eloquent ORM - это технология программирования, разработанная для упрощения работы с базами данных, предоставляя методы API для типичных операций (выборка, добавление, обновление, удаление и т. д.). Laravel предоставляет набор методов для создания и управления задачами, выполняемыми с помощью планировщика задач Cron [7].

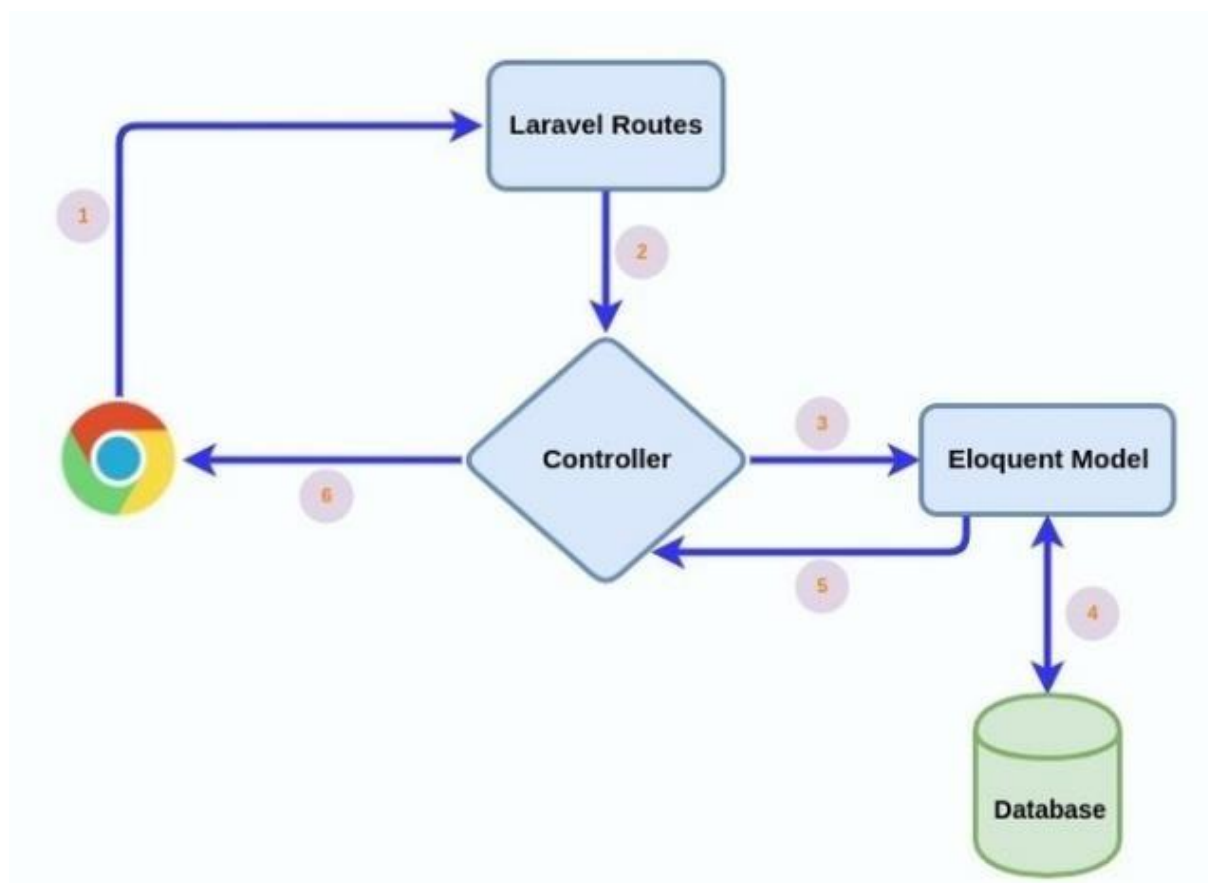


Рис. 2 Архитектура приложения на Laravel

К недостаткам Laravel можно отнести нарушение обратной совместимости между разными версиями Laravel; наличие синтаксического сахара, к которому разработчики очень быстро привыкают и забывают о том, как правильно писать чистые функции и запросы; нелогичное расположение файлов и каталогов в проекте; требуется хороший уровень владения английским языком: большинство полезных статей и официальной документации написано на английском языке, поэтому разработчики, не владеющие английским языком на уровне чтения технической документации, не смогут работать с этим фреймворком.

### 2.3. Spring MVC

Spring Framework - мощный фреймворк для разработки приложений. В более широком смысле Spring Framework - это хорошо разработанный инструмент, который поддерживает несколько веб-приложений, использующих Java в качестве языка программирования. Фреймворк был замечен на рынке благодаря модульности его базового функционала. Его можно разделить на разные модули, каждый из которых выполняет свою функцию.

Одной из важных особенностей Spring Framework является его легкость. Данный фреймворк очень легкий с точки зрения размера и функциональности благодаря реализации POJO, которая не требует наследования каких-либо классов и реализации каких-либо интерфейсов. Управление транзакциями используется для объединения нескольких API-интерфейсов управления транзакциями и координации транзакций объектов Java. Spring проектирует и управляет жизненным циклом и настройками объектов приложения. Внедрение зависимостей позволяет разрабатывать слабосвязанные приложения. Это значительно упрощает выполнение их модульного тестирования. Spring обеспечивает интеграцию фреймворков, решающих более сложные проблемы. Например, IBATIS, Hibernate, Toplink и т.д. [8].

Spring Web MVC, также известный как Spring MVC, является веб-средой Spring. Spring MVC позволяет создавать различные приложения: от небольших веб-сайтов до сложных веб-сервисов.

Вот так выглядит MVC в Spring (рис.3):

Model содержит данные, которые требуется отобразить на веб-странице. Однако данные полностью независимы от HTML, это простые объекты Java, из которых состоит приложение.

View будет HTML-шаблоном, который является каркасом для вашей HTML-страницы, написанной с помощью определенных библиотек шаблонов. Эти библиотеки позволяют включать заполнители в шаблоны, которые позволяют получить доступ к данным модели, например, имени пользователя.

Controller будет аннотированным методом @Controller, который отвечает на HTTP-запрос и знает, как преобразовать HTTP-запрос в объекты Java, а также объекты Java в ответ HTML.

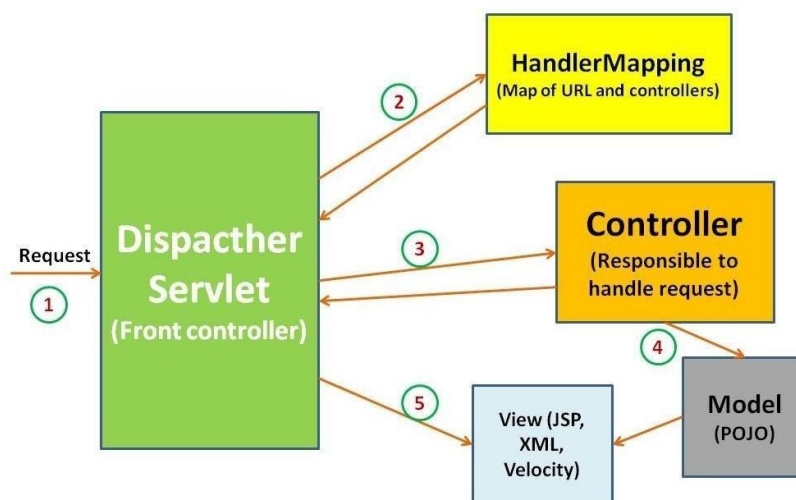


Рис. 3 Шаблон MVC в Spring MVC

К достоинствам Spring MVC однозначно можно отнести саму концепцию ориентированности на запросы: когда разработчик четко понимает, какие запросы и куда отправляются со стороны клиента, это приводит к более рациональному дизайну приложения, к более оптимизированному и быстрому

коду. Поддерживать приложение на Spring MVC легко, потому что все прозрачно. Нет зависимости от сторонних библиотек. Для создания визуальных компонентов и реализации определенной логики можно использовать библиотеки, известные всем веб-разработчикам, такие, как JQuery, Twitter Bootstrap и т.д. [9].

Главным недостатком является длительный срок разработки. Если вам нужно сделать систему в кратчайшие сроки, вы можете брать уже готовые компоненты и не беспокоиться о дизайне. Тогда вполне можно отказаться от Spring MVC.

## 2.4. Ruby on Rails

Ruby on Rails – это фреймворк для веб-разработки, написанный на языке программирования Ruby. Ruby постоянно развивается и окружен большим сообществом. Количество веб-сайтов, созданных с использованием этой платформы, быстро растет.

Философия Ruby on Rails включает два важных ведущих принципа. Don't Repeat Yourself - это принцип разработки программного обеспечения, который гласит, что «Каждая часть знания должна иметь единственное, непротиворечивое и авторитетное представление в рамках системы, авторитетное представление в системе». Convention Over Configuration: у Rails есть соглашения о наилучших способах выполнить множество задач в веб-приложении, и эти соглашения выставлены по умолчанию [10].

Разработка на Ruby on Rails имеет ряд преимуществ для проектов. Первое, что нужно учитывать при выборе правильной серверной среды - статус Open Source. Это означает, что Ruby on Rails может использоваться бесплатно. Поскольку Ruby on Rails является открытым исходным кодом, есть возможность внести свой вклад в развитие фреймворка. В Ruby on Rails огромное сообщество разработчиков. Множество программистов разрабатывает

огромное количество бесплатных дополнений, которые можно интегрировать в приложения.

Обширная экосистема делает Ruby on Rails превосходным по сравнению со многими другими фреймворками. RubyGems, сервис хостинга драгоценных камней сообщества Ruby, предоставляет доступ к тысячам различных «гемов», которые могут принимать форму дополнений, библиотек или фрагментов программного обеспечения. Драгоценные камни представляют собой готовые решения для различных проблем, которые упрощают процесс разработки. MVC (формат модель-представление-контроллер) - неотъемлемая часть инфраструктуры Ruby on Rails.(рис.4) Ruby on Rails допускает параллельную разработку и позволяет программистам ускорить данный процесс в три раза.

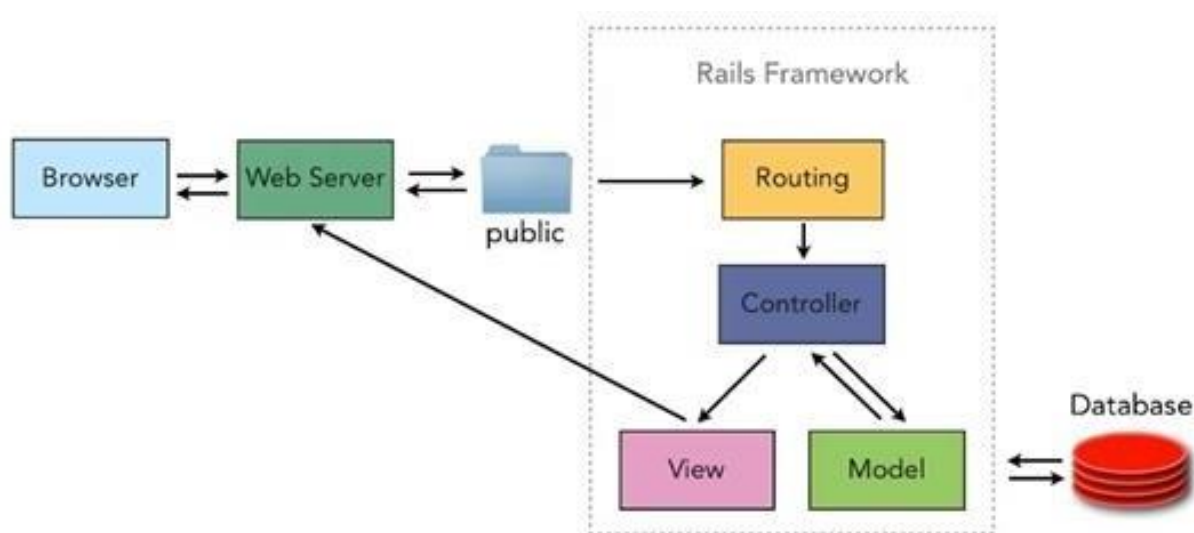


Рис. 4 Шаблон MVC в Ruby on Rails

Инфраструктура Rails предоставляет готовые корзины для разделения бизнес-логики приложения, поэтому компания-разработчик Ruby on Rails может сэкономить время за счет их использования. Разработчики Ruby on Rails могут использовать готовые части кода, что упрощает реализацию многих функций. В результате код приложения становится чистым и имеет высокую читаемость. Приложение, построенное на Ruby on Rails, может быть масштабировано для обработки тысяч запросов в секунду, отправленных

несколькими пользователями. В инфраструктуру встроены ориентированные на безопасность функции, которые делают приложения защищенными от SQL-инъекций и XSS-атак. Код Ruby самодокументируется. Это ускоряет процесс разработки, так как команде разработчиков не нужно выписывать отдельную документацию. В Ruby on Rails есть три среды по умолчанию, а именно: производство, разработка и тестирование. Весь цикл разработки оптимизирован. Есть возможность протестировать продукт, который разрабатывается на каждом этапе. Это приводит к меньшему количеству ошибок и ошибок, о которых вы должны знать и отлаживать.

Несмотря на то, что у Ruby on Rails есть много преимуществ, у него есть несколько недостатков, которые необходимо учитывать, прежде чем принимать решение о том, подходит ли он. Одним из главных недостатков является сложность поиска документации и ее недостаток. Скорость выполнения часто называют главным аргументом против Ruby on Rails. По сравнению со скоростью исполнения Ruby on Rails против Go, Ruby on Rails отстает. Тем не менее, когда мы рассматриваем Java-фреймворк Spring, Ruby on Rails побеждает в этой битве. И, наконец, не все хосты поддерживают Ruby on Rails. Фреймворк требует гораздо больше ресурсов, чем PHP, и не все хостинг-провайдеры могут обеспечить необходимые вычислительные мощности.

## ЗАКЛЮЧЕНИЕ

Таким образом, веб-фреймворк - это воплощение определенных шаблонов, интегрированного набора технологий, облегчающих разработку и поддержку веб-приложений. Чтобы выбрать фреймворк, нужно его проанализировать, изучить возможности и составить список того, что важно в зависимости от поставленной задачи. Необходимо иметь в виду, что список функций фреймворка - это лишь одна из его характеристик, и больше - не значит лучше.

При рассмотрении проекта с близким дедлайном правильным выбором для решения проблемы будет использование Django. Если вам нужно создать нетривиальный интерфейс, веб-приложение, которое будет работать очень быстро, вам следует выбрать Spring MVC. Нужно сделать выбор в пользу Laravel, если необходимо значительно упростить процесс создания веб-приложения. Если необходимо сконцентрироваться на методах и логике, создать компактное приложение, а не работать с большим объемом кода, то надо выбирать Ruby on Rails, но следует помнить, что язык программирования Ruby - не для новичков. Порог входа высок, поэтому нужно понимать принципы программирования и веб-разработки в целом.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Аверченков, В.И. Основы научного творчества [Электронный ресурс]: учеб. пособие/ В.И. Аверченков, Ю.А. Малахов. — Брянск: Брянский государственный технический университет, 2012. — 156 с.— Режим доступа: <http://www.iprbookshop.ru/7004>.
2. Рыжков, И.Б. Основы научных исследований и изобретательства. [Электронный ресурс]: учеб. пособие / И.Б. Рыжков. — СПб.: Лань, 2013. — 224 с. — Режим доступа: <http://e.lanbook.com/book/30202>.
3. Астанина, С.Ю. Научно-исследовательская работа студентов (современные требования, проблемы и их решения) [Электронный ресурс]: монография/ С.Ю. Астанина, Н.В. Шестак, Е.В. Чмыхова. — М.: Современная гуманитарная академия, 2012.— 156 с.— Режим доступа: <http://www.iprbookshop.ru/16934>.
4. Губарев, В.В. Квалификационные исследовательские работы [Электронный ресурс]: учеб. пособие/ В.В. Губарев, О.В. Казанская. — Новосибирск: Новосибирский государственный технический университет, 2014. — 80 с.— Режим доступа: <http://www.iprbookshop.ru/47691>.
5. Новиков, Ю.Н. Подготовка и защита бакалаврской работы, магистерской диссертации, дипломного проекта [Электронный ресурс] : учебное пособие / Ю.Н. Новиков. — 4-е изд., стер. — Санкт-Петербург : Лань, 2019. — 34 с. — URL: <https://e.lanbook.com/book/122187>.
6. Рябова К.М. Фреймворк Django: архитектура и возможности // Современные технологии: актуальные вопросы, достижения и инновации: сб. статей - Пенза, 2018. – С. 118-120
7. Stauffer M. Laravel: Up and Running: A Framework for Building Modern PHP Apps - O'Reilly Media, 2016 - 464 с.



8. Половинкин, А.И. Основы инженерного творчества. [Электронный ресурс]: учебное пособие / А.И. Половинкин.— СПб : Лань, 2019. — 364 с.— Режим доступа: <https://e.lanbook.com/book/123469>.
9. Рекомендации по написанию и оформлению курсовой работы, выпускной квалификационной работы и магистерской диссертации [Электронный ресурс]: учебно-методическое пособие/ Е.В. Зудина [и др.]. — Волгоград: Волгоградский государственный социально-педагогический университет, 2016. — 57 с.— Режим доступа: <http://www.iprbookshop.ru/57785>.
10. Фултон Х., Арко А. Путь Ruby - ДМК Пресс, 2016 - 666 с.