**СОДЕРЖАНИЕ**

[Введение 1](#_Toc138728154)

[1. Системы управления проектами 2](#_Toc138728155)

[2. Анализ методологий разработки 4](#_Toc138728156)

[3. Анализ существующих решений 5](#_Toc138728157)

[4. Анализ платформ разработки 8](#_Toc138728158)

[5. Анализ средств разработки серверной части системы 9](#_Toc138728159)

[5.1. Языки программирования 9](#_Toc138728160)

[5.2. Фреймворки 10](#_Toc138728161)

[5.3. Системы управления базами данных 11](#_Toc138728162)

[6. Анализ средств разработки клиентской части системы 12](#_Toc138728163)

[6.1. JS-фреймворки 14](#_Toc138728164)

[6.2. CSS-фреймворки 17](#_Toc138728165)

[6.3. Менеджеры состояний 19](#_Toc138728166)

[7. Реализация клиентской части веб-приложения 21](#_Toc138728167)

[8. Реализация серверной части веб-приложения 22](#_Toc138728168)

[Заключение 23](#_Toc138728169)

[Список литературы 24](#_Toc138728170)

Введение

В наше время бизнес разработки продуктов является одной из самых перспективных отраслей экономики. Благодаря постоянному развитию технологий и изменению потребительских предпочтений, появляются новые возможности для создания и продвижения инновационных продуктов на рынке.

Одним из ключевых факторов успеха в бизнесе разработки продуктов является умение быстро адаптироваться к изменяющимся условиям и требованиям рынка. Для этого необходимо постоянно улучшать процессы разработки, используя современные методологии и инструменты управления проектами.

Системы управления проектами играют важную роль в современном бизнесе, помогая организациям эффективно планировать, контролировать и реализовывать проекты различной сложности. Они помогают сокращать издержки и время на реализацию проектов [1], улучшать коммуникацию и сотрудничество между участниками проекта, управлять рисками и улучшать качество продуктов и услуг. Системы управления проектами также позволяют улучшить управление ресурсами, такими как время, бюджет и человеческие ресурсы.

Данные системы могут быть использованы в различных отраслях, таких как IT, строительство, производство, маркетинг, финансы и другие [2]. Они могут быть настроены для соответствия требованиям конкретной отрасли и размеру проекта, что делает их универсальными инструментами при разработке почти любых типов продуктов.

1. Системы управления проектами

Система управления проектами (Project Management System) – это комплекс программных и методологических средств, используемых для планирования, управления и контроля выполнения проектов [3]. Она включает в себя набор процессов, инструментов и техник, которые помогают организациям достигать своих целей, оптимизировать использование ресурсов и минимизировать риски.

Эффективное использование систем управления проектами позволяет достичь поставленных целей и задач в срок, снизить риски и улучшить качество продукта или услуги. Без подобных систем компании могут столкнуться с проблемами, такими как нехватка ресурсов, неопределенность целей и задач, превышение бюджета и неудовлетворительное качество работы.

При внедрении систем управления проектами в работу компании, зачастую у сотрудников возникают проблемы с ее использованием, помимо большого количества различных функций, такие системы могут иметь сложный интерфейс и множество различных элементов. Из-за трудностей в освоении и непонимании важности внедрения такой системы, у работников отсутствует желание и мотивация использовать ее в работе.

На основании данной проблемы было решено разработать систему управления проектами и задачами, которая обладала бы минимальным количеством основных функций, имела интуитивно понятный интерфейс, была проста в освоении, а также мотивировала сотрудников к обучению работе с подобными системами.

Разрабатываемая система предназначена для работников организаций, занимающихся разработкой различных продуктов. Благодаря ей, сотрудники смогут начать свое знакомство с системами управления проектами и начать успешно применять их в работе. С помощью системы они смогут назначать друг другу, выполнять, а также отслеживать прогресс выполнения задач. При этом пользователи будут разделены на две роли: работники, имеющие доступ к выполнению и просмотру всех задач на проекте, и менеджеры, также обладающие возможностью создавать и назначать задачи другим сотрудникам компании.

Данная выпускная квалификационная работа актуальна ввиду наличия в современном бизнесе разработки продуктов широкой потребности в информационной системе, которая способна облегчить интеграцию такого незаменимого инструмента, как системы управления проектами и задачами. Она поможет сотрудникам научиться применять данные системы в работе, а также получить опыт, который пригодится при работе с другими современными и популярными системами, обладающими большим количеством функций.

Объектом исследования является управление проектами с использованием СУП, а предметом – влияние систем управления проектами на эффективность работы компании.

Цель выпускной квалификационной работы – разработать информационную систему для управления проектами и задачами, выполняемыми сотрудниками организации.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Исследовать предметную область;
2. Провести анализ существующих решений в области систем управления проектами;
3. Сформировать список требований к информационной системе;
4. Провести анализ и выбрать используемые технологии разработки;
5. Спроектировать и реализовать серверную часть системы;
6. Спроектировать и реализовать клиентскую часть системы.
7. Анализ методологий разработки

Существует множество методологий управления проектами, каждая из которых представляет собой набор правил, процедур и методов, направленных на достижение целей проекта и управление его рисками. Рассмотрим несколько наиболее распространенных методологий [4]:

1. Agile – это методология разработки продуктов, которая основана на гибком и итеративном подходе к управлению проектами [5-6]. Она предполагает частые итерации и постоянное общение с заказчиком, чтобы быстро адаптироваться к изменениям требований и получать обратную связь. Agile используется в различных областях, таких как IT, маркетинг, производство и т. д.
2. Waterfall – это классическая последовательная методология, в которой каждая фаза проекта выполняется последовательно и завершается до перехода к следующей [7-8]. Она используется для проектов, где требования и цели четко определены и не меняются в процессе разработки. Преимуществом методологии Waterfall является ее предсказуемость и понятность, а недостатком – жесткость и невозможность быстро адаптироваться к изменениям.
3. Lean – это философия управления, которая ставит целью минимизацию потерь в процессах и повышение эффективности работы команды. Lean используется для управления процессами производства и предполагает постоянный анализ и оптимизацию процессов, чтобы устранить неэффективные шаги и увеличить производительность.
4. Scrum – это один из подходов Agile, который основан на коротких циклах разработки, называемых спринтами. Он используется для проектов, где цели и требования могут меняться в процессе разработки. Преимуществом методологии Scrum является ее гибкость и возможность быстрой адаптации. К недостаткам можно отнести необходимость строгой организации команды и выполнения спринтов в срок.
5. Kanban – это методология управления проектами, основанная на визуальном отображении рабочего процесса и управлении потоком работы. Она используется для улучшения эффективности проекта и сокращения времени, необходимого для выполнения задач [9-10]. Преимуществом методологии Kanban является ее простота и гибкость. Недостатком является отсутствие достаточного контроля над проектом и сложность визуализации проектов большой сложности.

Каждый из этих подходов имеет свои преимущества и недостатки, и может использоваться в различных областях и проектах [11]. Однако, в зависимости от проекта и задачи, может потребоваться комбинирование этих методологий для достижения наилучших результатов.

1. Анализ существующих решений

Существует множество систем управления проектами (СУП), которые помогают организовывать и контролировать выполнение проектов, снижать риски и повышать эффективность работы команды. Каждые из них имеют свои преимущества и недостатки, которые определяют, насколько они подходят для разработки продуктов в различных сферах.

Jira

Jira – это СУП, которая часто используется для разработки программного обеспечения (рис.1). Она позволяет организовывать и контролировать задачи, баги, выпуски, релизы и т. д. в рамках Agile-методологий разработки [12].

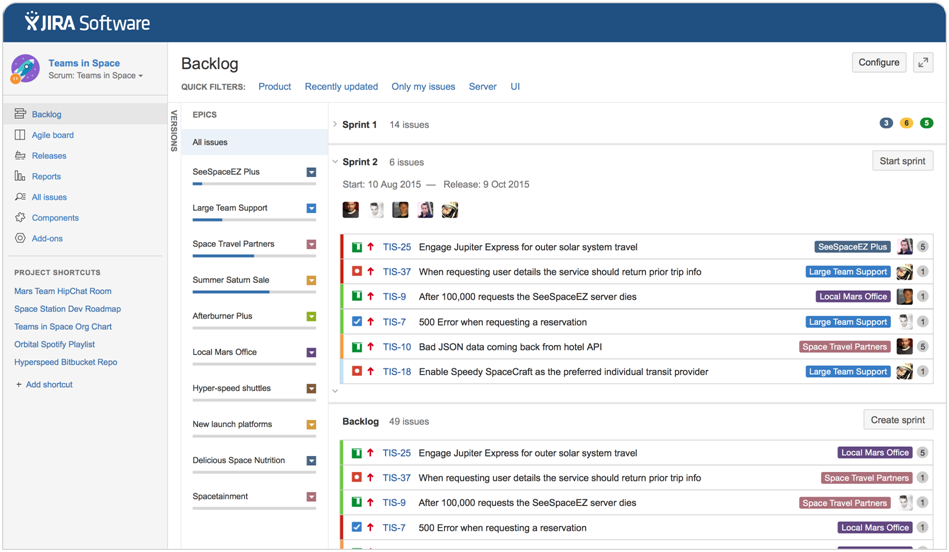


Рис.1 Интерфейс системы Jira

**Преимущества:**

1. Гибкие настройки, благодаря которым систему можно настроить под конкретные нужды команды;
2. Возможность отслеживать и визуализировать прогресс выполнения задач;
3. Возможность интеграции системы с другими инструментами разработки (Git, Confluence и т. д.).

**Недостатки:**

1. Может быть сложной в освоении из-за большого количества функций, что способно замедлить внедрение системы;
2. Проблемы с производительностью при большом количестве пользователей и задач;
3. Мало возможностей для автоматизации работы с системой.

Система лучше всего подходит для разработки программного обеспечения. Отлично справляется с большим количеством пользователей и задач. Также она имеет большое количество важных функций, способствующих комфортной работе всей команды, но из-за этого может быть трудна в освоении.

Trello

Trello – это простая и интуитивно понятная СУП (рис.2), которая широко применяется для управления проектами небольших команд или стартапов [13]. Она позволяет организовывать и контролировать задачи и процессы в удобном виде, используя карточки, списки и доски.

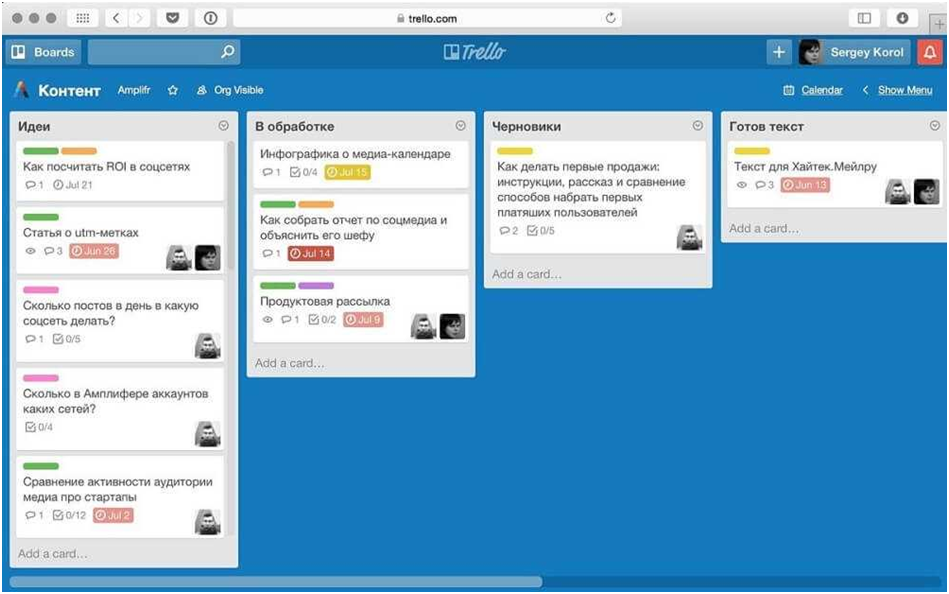


Рис.2 Интерфейс системы Trello

**Преимущества:**

1. Легкая и быстрая настройка, простота в использовании;
2. Гибкость, возможность подстроить систему под определенные потребности;
3. Возможность интеграции с другими инструментами разработки (GoogleDrive, Slack, Zapier).

**Недостатки:**

1. Ограниченный функционал системы, нет большого количества функций для комфортной работы;
2. Низкий уровень контроля и наблюдения за выполнением задач;
3. Трудности при работе с большим количеством проектов.

Система является очень простой в использовании, не обладая при этом большим количеством функций. Поэтому она наиболее подходит для совместной работы небольших команд, а также для личного пользования.

Asana

Asana – это СУП, которая позволяет организовывать и контролировать задачи и процессы на основе принципов Lean-методологии (рис.4). Она имеет широкий набор функций, таких как управление задачами, проектами, командами и т. д. [14].

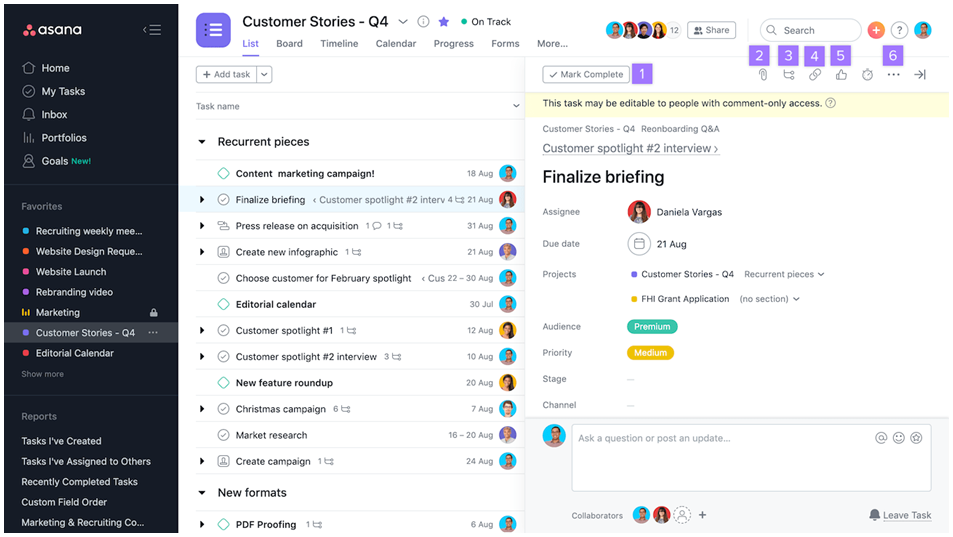


Рис.3 Интерфейс системы Asana

**Преимущества:**

1. Интуитивно понятный интерфейс;
2. Есть возможность добавления соисполнителей и наблюдателей в задачи;
3. Множество дополнительных удобных функций (календарь, напоминания, совместное использование документов и другие).

**Недостатки:**

1. Нет возможности добавление подзадач, что может быть неудобно при работе с крупными проектами;
2. Нет иерархии сотрудников;
3. Неудобства в работе с большим количеством задач.

Система оптимальна для команд и отделов среднего размера, подходит для работы в различных сферах. Имеет понятный и удобный интерфейс. При этом система не обладает функциями, предназначенными для разработки продуктов в отдельных сферах, например для разработки программного обеспечения.

1. Требования к информационной системе

На основе анализа существующих решений были сформированы требования к разрабатываемой системе. Благодаря тому, что рассмотренные системы являются самыми популярными и успешными в мире, необходимо сказать, что, реализовав некоторые их сильные стороны, а также исправив недочеты, можно разработать информационную систему, которая потенциально будет иметь схожие результаты.

* 1. Функциональные требования

Для того чтобы предоставлять пользователям основные возможности для планирования проектов и выполнения задач, разрабатываемая система должна выполнять следующие функции:

1. Авторизация и регистрация в системе;
2. Разделение ролей на работников и менеджеров для каждого проекта;
3. Создание проектов и добавление в них задач;
4. Смена статуса выполнения задачи;
5. Назначение задач конкретным пользователям в роли менеджера проекта;
6. Удаление задач из проекта в роли менеджера проекта;
7. Добавление и удаление пользователей из проекта в роли менеджера проекта;
8. Изменение роли конкретного пользователя;
9. Редактирование информации о пользователе и смена пароля, личный кабинет.
   1. Дополнительные требования

Помимо поддержки основных функций, система должна отвечать требованиям, необходимым для удобства пользователей:

* + 1. Интуитивно понятный интерфейс;
    2. Дополнительные функции(календарь, статистика, личные и групповые чаты и другие);
    3. Высокий уровень безопасности;
    4. Возможность комфортной работы с большим количеством функций;
    5. Гибкие настройки системы под определенные нужды компаний.

1. Анализ платформ разработки

Существуют различные типы программного обеспечения, которые выполняются на разных устройствах и используются для разных целей.

Десктопные приложения, также известные как приложения для настольных компьютеров, запускаются и работают на компьютере пользователя, без необходимости доступа к интернету. Такие приложения могут иметь более широкие функциональные возможности, чем мобильные и веб-приложения, и могут использоваться для более сложных задач, таких как обработка данных, проектирование и разработка, или для использования приложений, связанных с сенсорными устройствами и оборудованием, подключенными к компьютеру.

Мобильные приложения разработаны для использования на мобильных устройствах, таких как смартфоны и планшеты. Они часто имеют более узкую функциональность, чем десктопные приложения, и направлены на решение более специфических задач, таких как социальные медиа, банковские операции или организация задач и управление проектами. Важным фактором для мобильных приложений является оптимизация интерфейса и удобство использования на маленьких экранах.

Веб-приложения работают в браузере и требуют подключения к интернету. Они могут быть доступны через различные устройства и операционные системы, что делает их удобными для использования на любом устройстве с доступом в Интернет. Веб-приложения могут быть разработаны для разных целей, таких как электронная коммерция, облачное хранение данных или онлайн-сервисы, такие как электронная почта или календарь.

Каждый тип приложения имеет свои преимущества и недостатки, и выбор зависит от конкретных потребностей пользователя и задач, которые необходимо решить.

Системы управления проектами могут использоваться на различных устройствах, включая персональные компьютеры, ноутбуки, планшеты и смартфоны.

Для более сложных задач, таких как разработка проектного плана или анализ рисков, удобнее использовать более крупные устройства, такие как ноутбуки и настольные компьютеры, для обеспечения максимальной производительности и удобства работы с большим объемом информации.

Однако для управления проектом в реальном времени, мониторинга задач и обновления информации на ходу, мобильные устройства, такие как планшеты и смартфоны, могут быть более удобными. Также, мобильные устройства могут использоваться для быстрого доступа к системе управления проектами, когда нет возможности работать на стационарном компьютере или ноутбуке.

Благодаря своей универсальности веб-приложения наиболее подходят для разработки систем управления проектами. Такую систему пользователь сможет использовать на любом типе устройства, в любое время, и в любом удобном для него месте, где есть доступ в интернет.

1. Анализ средств разработки серверной части системы
   1. Языки программирования

В настоящее время существует огромное количество различных языков программирования. Каждый из них обладает своим синтаксисом, принципами разработки и устройством, благодаря этим особенностям, разные языки используются для разных типов задач и при разработке разных приложений.

Основные языки разработки серверной части веб-приложений

Только некоторые из языков программирования подходят для разработки серверной части веб-приложений, из них самыми популярными являются:

1. JavaScript – это язык программирования, который может быть использован как для разработки клиентской, так и серверной частей веб-приложения. Для серверной разработки на JavaScript используются фреймворки и среды выполнения, такие как Node.js, Express.js, Nest.js и другие.
2. Python – простой и популярный язык программирования, который имеет широкие возможности для разработки серверных приложений. Для серверной разработки на Python широко используются различные фреймворки, такие как Django, Flask, Pyramid и Tornado.
3. Java – популярный и мощный язык программирования, используемый для разработки масштабных серверных приложений. Для веб-разработки на Java широко применяются фреймворки, такие как Spring и JavaServer Faces (JSF).
4. PHP – язык программирования, наиболее распространенный для разработки веб-приложений. Для серверной разработки на PHP часто используются такие фреймворки, как Laravel, Symfony и CodeIgniter.

Каждый из перечисленных языков имеет все функции и предоставляет возможности для комфортной серверной разработки. Выбирая один из данных языков, стоит учитывать их особенности, а также отталкиваться от необходимостей и приоритетов при проектировании приложения.

Выбор языка программирования для серверной части системы

Исходя из статистических данных [https://techrocks.ru/2018/07/29/programming-languages-popularity-2018/], в настоящее время наибольшей популярностью пользуется язык программирования Java (рис. 4). Это означает то, что он больше остальных подходит для разработки серверной части. Помимо этого, Java обладает большим количеством библиотек и фреймворков, понятным синтаксисом и повышенной безопасностью, а также предоставляет возможность для масштабирования приложений.

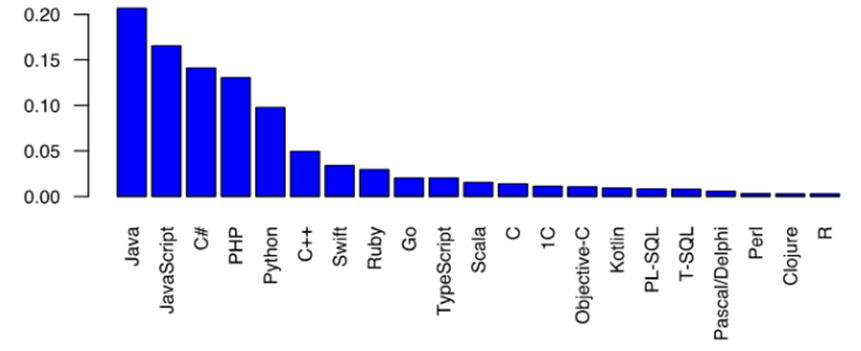


Рис.4 Популярность языков программирования по годам

Приложения, написанные на этом языке программирования будут работать одинаково на любой операционной системе или платформе, поддерживающей виртуальную машину Java (JVM), без необходимости каких-либо значительных модификаций. Также его преимуществом является высокая надежность, благодаря предоставляемым механизмам обработки ошибок и управления исключениями, которые необходимы для создания стабильных и отказоустойчивых приложений.

* 1. Фреймворки

Разработку серверной части веб-приложений невозможно представить без использования фреймворков, которые существенно упрощают разработку и сокращают ее время. Данных фреймворков существует огромное множество, при этом все они имеют свои особенности и различные наборы инструментов и библиотек.

Spring

Spring – мощный фреймворк на языке Java для разработки серверных приложений. Он предоставляет множество модулей, таких как Spring MVC, Spring Boot и Spring Data, для облегчения разработки и интеграции.

Node.js

Node.js – среда выполнения JavaScript и Express.js – легкий и гибкий веб-фреймворк для Node.js. Позволяют разрабатывать мощные API и веб-приложения.

Django

Django – мощный фреймворк на языке Python для разработки веб-приложений. Он обеспечивает высокую производительность, безопасность и включает в себя множество функций «из коробки».

Laravel

Laravel – популярный фреймворк на языке PHP для разработки веб-приложений. Он предлагает элегантный синтаксис, расширяемость и широкий набор функций, таких как маршрутизация, работа с базами данных и аутентификация.

Выбор фреймворка для серверной части системы

Тремя самыми распространенными фреймворками для разработки серверной части веб-приложений на 2023 год являются Laravel, Django и Spring (рис.5). И ввиду того, что ранее для разработки системы было решено использовать язык программирования Java, наиболее подходящим вариантом для выбора фреймворка является Spring.

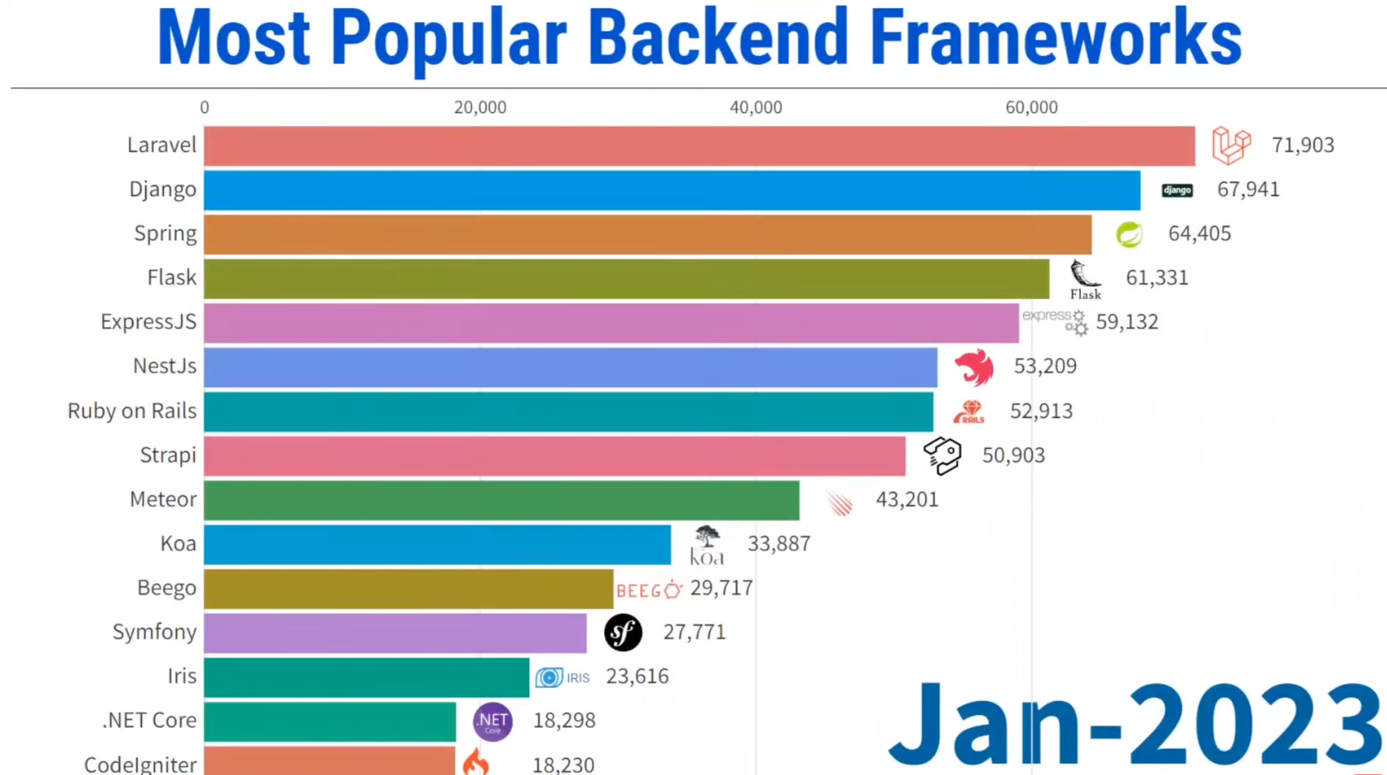


Рис.5 Статистика использования фреймворков при серверной разработке

Фреймворк Spring предлагает простую и интуитивно понятную модель программирования, предоставляет множество готовых компонентов, которые можно использовать для обработки типичных задач, построен на основе модульной архитектуры, что позволяет использовать только необходимые модули и функциональность в приложении. Также он дает разработчикам возможность интеграции с множеством других популярных технологий и фреймворков, таких как Hibernate для работы с базами данных, Spring MVC для создания веб-приложений, Spring Security для обеспечения безопасности и многие другие. Это делает Spring универсальным фреймворком для разработки комплексных и масштабируемых приложений.

* 1. Системы управления базами данных

При разработке веб-приложений часто требуется использовать Системы Управления Базами Данных (СУБД) для хранения и управления данными.

Система управления базами данных (СУБД) ‑ это программное обеспечение, разработанное для создания, хранения, организации, управления и обработки больших объемов данных.

СУБД играют важную роль в различных сферах, включая веб-разработку, корпоративные приложения, аналитику данных, научные исследования и многое другое. Они обеспечивают эффективное хранение и доступ к данным, обеспечивая целостность, безопасность и удобство работы с информацией.

MySQL

MySQL – одна из самых популярных открытых реляционных СУБД. Она обеспечивает высокую производительность, надежность и поддерживает множество функций, таких как транзакции, схемы безопасности и репликация.

Основными характеристиками и возможностями MySQL являются:

* Реляционная модель данных, где данные организованы в таблицы с рядами и столбцами. Это обеспечивает структурированное хранение и эффективные операции с данными.
* Поддержка стандартного языка запросов SQL (Structured Query Language) для выполнения операций чтения, записи, обновления и удаления данных. SQL обеспечивает простой и мощный способ взаимодействия с базой данных.
* Возможности масштабирования и обработки больших объемов данных. СУБД может работать на одном сервере или в кластере серверов для обеспечения высокой производительности и отказоустойчивости.
* Поддержка транзакций, что позволяет выполнение группы операций как одну атомарную и неделимую единицу. Это гарантирует целостность данных и обеспечивает возможность отката изменений в случае ошибок.
* Поддержка различных типы хранилищ данных, включая InnoDB, MyISAM, MEMORY и другие. Каждый тип хранилища имеет свои особенности и предназначен для определенных сценариев использования.
* Широкий спектр функций, таких как поддержка субзапросов, представлений, хранимых процедур, триггеров, индексов, агрегатных функций и многое другое. Это делает MySQL мощным инструментом для работы с данными.
* Поддержка различных платформ, включая Windows, Linux, macOS и другие операционные системы. Есть возможность интеграции с различными языками программирования, такими как PHP, Python, Java и другие.

MySQL является одной из самых распространенных СУБД и широко применяется веб-приложениями, корпоративными системами, а также во множестве других сценариев, где требуется эффективное хранение и управление данными.

MongoDB

MongoDB – гибкая и масштабируемая документоориентированная СУБД, использующая формат BSON (расширенный JSON). Она позволяет хранить и обрабатывать документы без жесткой схемы и хорошо подходит для разработки современных веб-приложений.

Основными характеристиками и возможностями MongoDB являются:

* Использование модели данных, основанной на документах. Вместо использования таблиц и строк, данные хранятся в гибком формате JSON-подобных документов, что позволяет более естественно представлять данные, их связи и иерархию.
* Простое горизонтальное масштабирование путем добавления новых серверов в кластер. Это позволяет обрабатывать большие объемы данных и обеспечивает высокую доступность и отказоустойчивость.
* Гибкие схемы данных, благодаря которым каждый документ в коллекции может иметь различную структуру. Это позволяет легко изменять схему данных по мере развития приложения без необходимости миграции данных.
* Мощный и гибкий язык запросов, называемый MongoDB Query Language (MQL). Он позволяет выполнять сложные запросы, фильтровать, сортировать и агрегировать данные, а также поддерживает географические запросы.
* Поддержка различных типов индексов, включая одиночные и составные индексы, текстовый поиск и географические индексы. Индексы позволяют ускорить выполнение запросов и повысить производительность базы данных.
* Механизмы для репликации данных, что обеспечивает высокую доступность и отказоустойчивость. Также возможно горизонтальное шардинг, разделение данных на несколько серверов для более эффективного масштабирования.
* Поддержка ACID-транзакции внутри одной коллекции. Это позволяет выполнять группу операций как одну атомарную и неделимую единицу, обеспечивая целостность данных.

MongoDB широко применяется в различных приложениях, таких как веб-разработка, аналитика, хранение логов, управление контентом и другие сценарии, где требуется гибкое хранение и обработка больших объемов структурированных и неструктурированных данных.

PostgreSQL

PostgreSQL – мощная открытая реляционная СУБД с широкими возможностями. Она является открытым программным обеспечением и предлагает широкий набор функций для хранения, управления и анализа данных.

Система управления базами данных PostgreSQL обладает следующими основными особенностями:

* Реляционная модель данных, которая обеспечивает структурированное хранение данных и поддержку связей между таблицами.
* Полное соответствие стандарту SQL и предлагает расширенный набор возможностей языка запросов. Он поддерживает сложные запросы, операции соединения таблиц, агрегатные функции, оконные функции и многое другое.
* Поддержка ACID-транзакций (атомарность, согласованность, изолированность, долговечность), что гарантирует надежность и целостность данных. Наличие механизмов проверки ограничений и триггеров для обеспечения бизнес-правил и автоматического выполнения операций.
* Поддержка горизонтального масштабирования путем репликации данных и создания кластеров. Оптимизация запросов, индексы, партиционирование таблиц и другие механизмы для обеспечения высокой производительности.
* Возможность создания пользовательских функций, типов данных и расширений. Это позволяет разработчикам расширять функциональность и создавать собственные средства для обработки данных.
* Встроенную поддержка для работы с JSON-данными, позволяющая хранить и выполнять операции с JSON-структурами. Наличие полнотекстового поиска, что обеспечивает мощные возможности поиска и анализа текстовых данных.
* Механизмы аутентификации и авторизации для защиты данных. Поддержка шифрования, ролевой модели доступа, SSL-соединений и другие средств безопасности.

PostgreSQL является одной из самых надежных и популярных систем управления базами данных. Она широко применяется в различных областях, включая корпоративные приложения, веб-разработку, геоинформационные системы, аналитику данных и многое другое.

Выбор СУБД для серверной части системы

Таблица 1 – Преимущества и недостатки СУБД

|  |  |  |  |
| --- | --- | --- | --- |
|  | **MySQL** | **MongoDB** | **PostgreSQL** |
| **Преимущества** | Простота использования и настройки | Гибкая схема данных | Богатый набор функций SQL |
| Хорошая производительность | Горизонтальное масштабирование | Поддержка ACID-транзакций |
| Широкая поддержка языков программирования | Высокая скорость записи и чтения | Большой набор расширений и пользовательских функций |
| Хорошая поддержка индексов | Поддержка JSON и географических запросов | Высокая надежность и целостность данных |
| **Недостатки** | Ограниченная горизонтальная масштабируемость | Нет поддержки транзакций между коллекциями | Больший объем памяти для работы с данными |
| Ограниченная гибкость схемы данных | Ограниченная поддержка соединений между коллекциями | Сложнее в настройке и управлении |
| Ограниченная поддержка JSON | Более высокое потребление памяти | Меньшая производительность при больших объемах данных |

1. Анализ средств разработки клиентской части системы
   1. Языки программирования

Клиентская часть веб-приложений отвечает за взаимодействие с пользователем и отображение данных в браузере.

Основой для разработки веб-приложений является язык разметки HTML (HyperText Markup Language), необходимый для создания структуры и содержимого веб-страниц, с помощью различных элементов и тегов, таких как заголовки, параграфы, списки, изображения и другие. Для оформления и стилизации веб-страниц используется CSS (Cascading Style Sheets), он позволяет создавать красивые и привлекательные пользовательские интерфейсы, а также обеспечивает адаптивность и отзывчивость веб-приложений. Без данных инструментов разработка веб-приложения невозможна, при этом они используются только вместе с одним из языков программирования.

Для разработки клиентской части веб-приложений на данный момент существуют два основных языка программирования: JavaScript и TypeScript. Эти языки во многом похожи, но при этом обладают некоторыми существенными отличиями.

JavaScript

JavaScript – мультипарадигменный язык программирования для клиентской части веб-приложений, позволяющий создавать интерактивные элементы, управлять динамическим содержимым веб-страницы, выполнять асинхронные запросы к серверу и многое другое. Он поддерживается всеми современными браузерами. JavaScript также используется во многих фреймворках и библиотеках.

TypeScript

TypeScript – язык программирования, основанный на JavaScript, но с добавлением статической типизации. Он предоставляет разработчикам инструменты для разработки крупномасштабных приложений, облегчая обнаружение ошибок на этапе разработки и повышая поддержку кода. TypeScript компилируется в JavaScript и может использоваться вместе с различными фреймворками.

Выбор языка программирования для клиентской части системы

Несмотря на то, что TypeScript основан на JavaScript, они имеют некоторые отличия, такие как:

* Синтаксис и типизация. JavaScript является динамически типизированным языком программирования, что означает, что типы переменных определяются автоматически во время выполнения программы. Он не требует явного указания типов данных. TypeScript, с другой стороны, является статически типизированным языком, что означает, что типы переменных определяются на этапе компиляции. Он добавляет статическую типизацию к JavaScript, позволяя разработчикам указывать типы переменных, параметров функций и возвращаемых значений.
* Компиляция. JavaScript выполняется непосредственно в браузере или на сервере без предварительной компиляции. Он интерпретируется и выполняется по мере загрузки веб-страницы. TypeScript, как язык со статической типизацией, требует предварительной компиляции. Исходный код TypeScript компилируется в JavaScript перед его выполнением. Это означает, что перед использованием кода TypeScript необходимо его скомпилировать в JavaScript.
* Расширенные возможности. TypeScript предлагает дополнительные функции, которых нет в JavaScript. Например, TypeScript поддерживает статическую типизацию, что может помочь обнаруживать ошибки на этапе компиляции и повышает надежность кода. TypeScript также включает возможности использования классов, интерфейсов, перечислений и других объектно-ориентированных конструкций, которые упрощают разработку и поддержку сложных проектов.
* Совместимость. Поскольку TypeScript является надмножеством JavaScript, весь действительный код JavaScript является допустимым кодом TypeScript. Это означает, что вы можете постепенно внедрять TypeScript в существующий JavaScript-код и использовать их вместе.
* Экосистема и сообщество. JavaScript имеет обширную экосистему с множеством библиотек, фреймворков и инструментов, доступных для разработчиков. Он является одним из самых популярных языков программирования веб-разработки. TypeScript также имеет широкую поддержку и активное сообщество разработчиков. Он интегрируется с различными инструментами разработки, такими как Visual Studio Code, и широко используется в крупных проектах.

Благодаря их отличиям, можно выделить плюсы и минусы для обоих языков программирования.

Таблица 2 – Преимущества и недостатки языков клиентской разработки

|  |  |  |
| --- | --- | --- |
| **Язык программирования** | **JavaScript** | **TypeScript** |
| **Преимущества** | Универсальность; Большая экосистема; Простота изучения. | Статическая типизация; Улучшенная разработка; Совместимость с JavaScript. |
| **Недостатки** | Динамическая типизация; Отсутствие строгих правил; Проблемы с безопасностью. | Необходимость компиляции; Ограничения экосистемы; Сложность изучения. |

TypeScript предоставляет дополнительные функции и инструменты, которые могут улучшить процесс разработки и поддержки больших проектов. Однако JavaScript является самым популярным языком веб-разработки (рис. 6), благодаря чему имеет большую экосистему и сообщество. При этом он является простым в изучении, надежным и проверенным временем инструментов, в отличии TypeScript, языка программирования, сложного из-за большого количества дополнительных функций и появившегося относительно недавно.

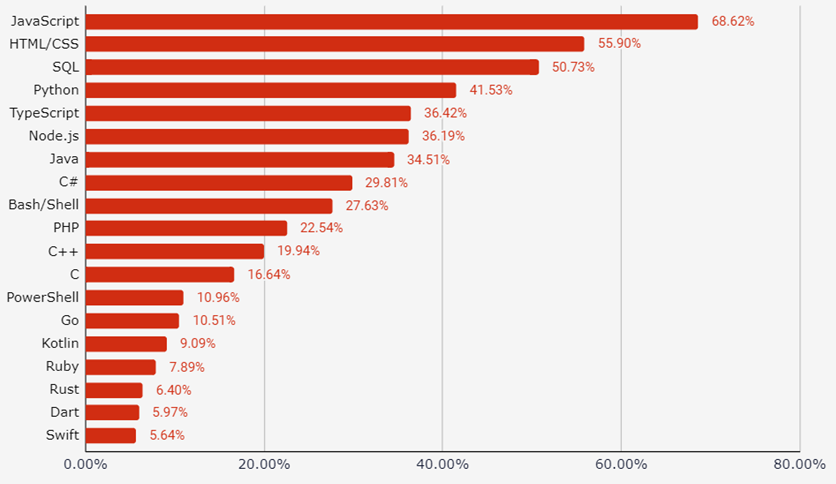


Рис.6 Использование языков программирования при разработке веб-приложений

* 1. JS-фреймворки

JavaScript-фреймворк – это набор предопределенных структур, библиотек и инструментов, созданных на основе языка JavaScript, с целью упростить разработку веб-приложений. Фреймворк предоставляет разработчикам готовые решения и стандартизированные подходы к разработке, чтобы ускорить процесс и облегчить задачи, связанные с созданием сложных веб-приложений.

Фреймворки обычно предлагают ряд функциональных возможностей и инструментов, таких как:

* Архитектура приложения. Фреймворки обычно предлагают определенную архитектуру приложения, которая помогает организовать код и структуру приложения. Например, многие фреймворки используют концепцию компонентов или модулей для организации кода.
* Маршрутизация. Фреймворки часто предлагают механизм маршрутизации, который облегчает управление маршрутами и навигацией в приложении. Это позволяет создавать SPA (одностраничные приложения), где содержимое обновляется динамически без перезагрузки страницы.
* Управление состоянием. Многие фреймворки предлагают инструменты для управления состоянием приложения. Они обеспечивают механизмы для хранения, изменения и обновления данных приложения на разных уровнях.
* Визуальные компоненты. Фреймворки предоставляют набор предопределенных компонентов и UI-элементов, которые можно использовать для создания пользовательского интерфейса. Это помогает ускорить разработку и обеспечить единообразный внешний вид и поведение интерфейса.
* Инструменты для тестирования и отладки. Многие фреймворки предлагают инструменты для автоматизации тестирования приложения и обнаружения и исправления ошибок.

JavaScript имеет множество фреймворков, которые облегчают разработку веб-приложений. Из них наиболее широкий спрос среди разработчиков имеют React, Angular и Vue. При этом данные фреймворки сильно различаются по многим аспектам, в том числе своими преимуществами и недостатками.

React

React (или React.js) – это библиотека JavaScript, разработанная компанией Facebook, которая используется для создания пользовательских интерфейсов (UI) веб-приложений. Она позволяет разработчикам строить компоненты, которые представляют собой небольшие, изолированные части пользовательского интерфейса, и затем объединять их в сложные приложения.

React использует концепцию виртуального DOM (Document Object Model), что означает, что он создает в памяти представление пользовательского интерфейса и эффективно обновляет только измененные части, минимизируя количество операций на реальном DOM. Это делает React очень эффективным и быстрым в работе.

React также предоставляет механизм управления состоянием приложения, который позволяет легко отслеживать и обновлять данные во время выполнения приложения. Он позволяет создавать динамические интерфейсы, которые реагируют на пользовательские действия или изменения данных.

React можно использовать для разработки одностраничных приложений (SPA), мобильных приложений, десктопных приложений и других типов веб-приложений. Он широко используется в индустрии разработки веб-приложений и имеет большую и активную сообщество разработчиков.

Angular

Angular (или Angular.js) – это фреймворк для разработки веб-приложений, разработанный компанией Google. Он предоставляет инструменты и функциональность для создания эффективных и масштабируемых одностраничных приложений (SPA).

Angular использует шаблонный подход для создания пользовательского интерфейса, основанный на расширении HTML с использованием директив и связывания данных (data binding). Он позволяет разработчикам создавать компоненты, которые представляют отдельные части пользовательского интерфейса, и объединять их для построения сложных приложений.

Основные особенности Angular:

* Компонентный подход. Приложение строится из набора компонентов, которые взаимодействуют друг с другом. Каждый компонент содержит свой HTML-шаблон, логику и стили.
* Двустороннее связывание данных. Angular предоставляет механизм двусторонней связи данных, который автоматически обновляет пользовательский интерфейс при изменении данных и наоборот.
* Внедрение зависимостей. Angular имеет встроенную систему внедрения зависимостей (Dependency Injection), которая упрощает управление зависимостями и создание тестируемого кода.
* Роутинг. Angular предоставляет возможность маршрутизации и навигации между различными компонентами и представлениями внутри приложения.
* Расширяемость. Angular обладает модульной архитектурой, которая позволяет добавлять сторонние модули и библиотеки для расширения функциональности приложения.

Angular написан на языке TypeScript, который предоставляет статическую типизацию и другие возможности, упрощающие разработку и обслуживание приложений.

Vue

Vue.js (или просто Vue) – это прогрессивный JavaScript-фреймворк для разработки пользовательских интерфейсов. Он является открытым проектом и разрабатывается активным сообществом разработчиков. Vue позволяет создавать интерактивные веб-приложения, аналогично React и Angular.

Основные особенности Vue:

* Простота изучения. Vue обладает простым и интуитивным синтаксисом, что делает его легко изучаемым для новичков и быстро применимым в проектах.
* Компонентная архитектура. Vue основан на компонентах, которые могут быть созданы и переиспользованы для построения пользовательского интерфейса. Компоненты Vue включают шаблоны HTML, стили CSS и логику JavaScript.
* Реактивность. Vue обеспечивает реактивное поведение данных, что означает, что изменения данных автоматически отражаются в пользовательском интерфейсе без явного обновления. Это делает код более понятным и удобным для разработки.
* Директивы. Vue предоставляет множество встроенных директив, которые позволяют легко управлять поведением компонентов. Директивы позволяют добавлять интерактивность и реагировать на пользовательские события.
* Виртуальный DOM. Аналогично React, Vue использует виртуальный DOM для эффективного обновления только измененных частей пользовательского интерфейса.
* Расширяемость. Vue предоставляет возможность создавать собственные плагины и расширения для добавления дополнительных функций и интеграции с другими библиотеками.

Vue можно использовать в различных проектах – от небольших интерактивных компонентов до больших масштабируемых приложений. Он также имеет официальные пакеты и инструменты для разработки мобильных приложений (Vue Native) и десктопных приложений (Vue.js + Electron).

Выбор JS-фреймворка для клиентской части системы

По статистике, больше всего веб-приложений в настоящее время разрабатывается с использованием JavaScript фреймворка React, затем идет Angular, Vue и остальные (рис. 6).

Рис.6 Использование фреймворков при клиентской разработке

React признан лучшим фреймворком для клиентской части, он обеспечивает удобную разработку, подходит для создания больших и масштабируемых веб-приложений, в отличии от Vue, но при этом не такой сложный в изучении, как Angular. Благодаря этому React является идеальным выбором для разработки клиентской части веб-приложений.

* 1. CSS-фреймворки

CSS-фреймворк (Cascading Style Sheets Framework) – это набор предварительно написанных стилей CSS, который предоставляет готовые компоненты и соглашения о стилях для упрощения разработки веб-интерфейсов. Он облегчает создание современного и отзывчивого дизайна, а также повышает эффективность работы разработчиков, предоставляя готовые решения для общих задач.

Данные фреймворки обычно содержат набор классов CSS, которые можно применять к HTML-элементам для быстрой настройки их внешнего вида и расположения. Эти классы предопределены и стилизованы заранее, позволяя разработчикам создавать согласованный и профессиональный дизайн, не тратя много времени на написание стилей с нуля.

Также такие фреймворки могут быть полезными инструментами для быстрой разработки и обеспечения консистентности дизайна, особенно если у вас ограниченное время или опыт веб-разработки. Однако, при использовании фреймворков, важно понимать их особенности и правильно применять их классы и компоненты в соответствии с требованиями вашего проекта.

Существует большое количество различных CSS-фреймворков. Каждый предлагает свои уникальные возможности, набор компонентов и стилей, а также подходы к организации кода и адаптивности.

Bootstrap

Bootstrap – это один из самых популярных и широко используемых CSS-фреймворков. Он предоставляет множество готовых компонентов, таких как кнопки, формы, навигационные меню, карусели и многое другое. Bootstrap также имеет мощную сеточную систему, которая помогает создавать адаптивные веб-страницы. Он легко настраивается и расширяется, и обладает обширной документацией и активным сообществом разработчиков.

Foundation

Foundation – это гибкий CSS-фреймворк, который также предоставляет набор готовых компонентов и сеточную систему. Он изначально разработан для создания адаптивных и мобильных веб-приложений. Foundation обладает модульной архитектурой, которая позволяет разработчикам выбирать и использовать только необходимые компоненты. Он также предлагает набор гибких тем и инструментов для настройки внешнего вида и поведения компонентов.

Bulma

Bulma – легкий и простой в использовании CSS-фреймворк. Он предоставляет готовые классы для создания различных компонентов и макетов, таких как колонки, навигация, формы и другие элементы интерфейса. Bulma обладает простым и интуитивным синтаксисом, а также хорошо настраивается с помощью переменных Sass. Он не требует JavaScript для работы, что может быть преимуществом в некоторых случаях.

Material-UI

Material-UI – это CSS-фреймворк, основанный на принципах дизайна Material Design от Google. Он предоставляет готовые компоненты и стили, соответствующие спецификации Material Design. Material-UI также обладает возможностями для создания анимации и эффектов. Он интегрируется хорошо с React, но также может использоваться и с другими фреймворками или библиотеками JavaScript. Material-UI предлагает множество настраиваемых тем и стилей, а также инструменты для создания собственных компонентов.

Выбор CSS-фреймворка для клиентской части системы

Все перечисленные фреймворки являются современными средствами для разработки пользовательских интерфейсов веб-приложений. Но при этом, каждый из них имеет свои сильные и слабые стороны.

Таблица 3 – Преимущества и недостатки CSS-фреймворков

|  |  |  |
| --- | --- | --- |
| **Фреймворк** | **Преимущества** | **Недостатки** |
| **Bootstrap** | Огромное сообщество и обширная документация; Большой набор готовых компонентов и стилей; Гибкая сеточная система; Легкая настройка. | Шаблонность; Большой размер и сложность. |
| **Foundation** | Гибкость и модульность; Адаптивность; Мощная сеточная система. | Мало готовых компонентов; Сложная настройка. |
| **Bulma** | Простота использования; Интуитивный синтаксис; Настраивается с помощью Sass; Не требует JavaScript. | Мало готовых компонентов; Относительно небольшое сообщество. |
| **Material-UI** | Основан на Material Design; Широкий набор готовых компонентов; Хорошая поддержка React и других JS-фреймворков; Множество настраиваемых тем и стилей. | Специфичность; Относительно сложная настройка. |

Для разработки веб-приложения с использованием JavaScript фреймворка React наиболее подходящим является Material-UI. Он обладает улучшенной поддержкой JS-фреймворков, что делает разработку более удобной. Также он предоставляет возможность тонкой настройки стилей всех компонентов, позволяя разрабатывать привлекательные и понятные пользовательские интерфейсы (рис 7.).

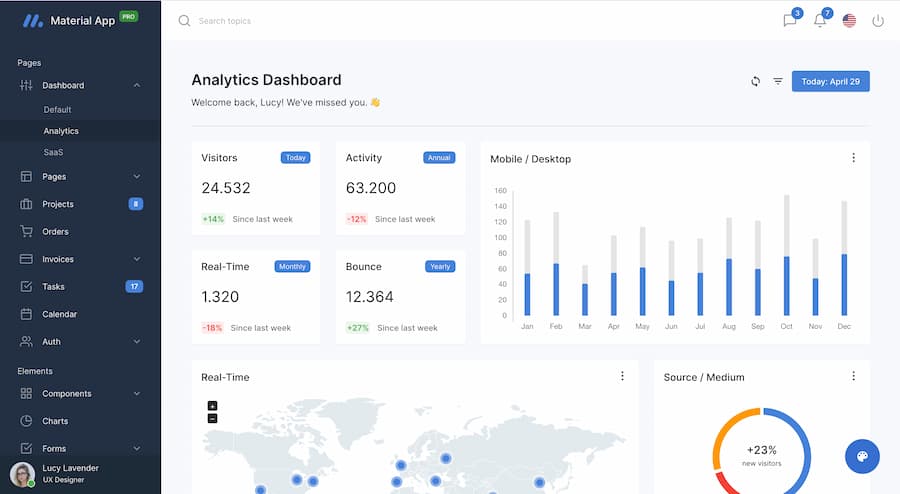


Рис.7 Пример интерфейса, разработанного с помощью MUI

* 1. Менеджеры состояний

Менеджер состояний (State Manager) в веб-приложениях – это инструмент или библиотека, которая помогает организовать и управлять состоянием данных приложения. Состояние обычно включает в себя информацию, которая может изменяться в процессе работы приложения, такую как пользовательский ввод, получение данных с сервера или изменение состояния компонентов интерфейса.

Менеджеры состояний обеспечивают централизованное хранение и управление состоянием приложения, что упрощает синхронизацию данных между различными компонентами. Они предлагают способы для обновления, доступа и отслеживания состояния, а также обрабатывают изменения состояния и обновление интерфейса при необходимости.

Некоторые из популярных менеджеров состояний в веб-разработке включают:

Redux

Redux – это популярный менеджер состояний для JavaScript-приложений. Он работает на основе принципа однонаправленного потока данных и централизованного хранилища состояния. Redux использует концепцию «actions» (действия), «reducers» (редьюсеры) и «store» (хранилище) для управления состоянием и обновлением интерфейса.

MobX

MobX – менеджер состояний, который обеспечивает реактивное управление состоянием. Он использует наблюдение (observables) и реакции (reactions) для автоматического отслеживания изменений состояния и обновления соответствующих компонентов. MobX обладает простым синтаксисом и удобным API.

Recoil

Recoil – это библиотека для управления состоянием, разработанная Facebook. Она предоставляет простой и декларативный способ определения и использования атомов (atoms) – независимых единиц состояния. Recoil обеспечивает автоматическую отслеживаемость зависимостей и эффективное обновление только необходимых компонентов при изменении состояния.

Выбор менеджера состояний для клиентской части системы

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Redux** | **MobX** | **Recoil** |
| Преимущества | Предсказуемость и контроль состояния; Большая экосистема и поддержка сообщества; Удобная инструментальная поддержка; Средства для отладки и мониторинга; | Простота использования; Реактивное программирование; Гибкость в изменении состояния; Меньше шаблонного кода для настройки; Быстрая разработка и прототипирование. | Интуитивная модель управления состоянием; Хорошая производительность; Асинхронная обработка состояния; Легкое масштабирование и композиция. |
| Недостатки | Большее количество кода для настройки; Более сложная структура; Больший порог входа. | Большее потребление ресурсов; Более сложная структура; Больший порог входа для новичков. | Меньшая экосистема и сообщество; Новый проект с меньшей известностью. |

1. Реализация клиентской части веб-приложения
2. Реализация серверной части веб-приложения

Заключение

Системы управления проектами являются невероятно важным инструментом при разработке продуктов. Они помогают планировать и отслеживать прогресс разработки, улучшают качество продукта, а также способны существенно сокращать все издержки.

Существуют различные методологии, на которых основаны данные системы. Каждая из них имеет свои преимущества и недостатки. Выбор методологии зависит от конкретных целей и задач компании, при этом часто используются их комбинации. Было проанализировано большинство популярных методологий, для ознакомления с предметной областью.

На основе методологий управления проектами в наше время разрабатывается большое количество различных систем. Были рассмотрены и проанализированы самые популярные из них. Некоторые из систем управления проектами являются простыми в использовании и универсальными для разных сфер разработки, другие же обладают большим количеством функций, заточенных под определенный тип продукта. Были выявлены слабые и сильные стороны каждой системы. Выделенные плюсы и минусы решений могут быть использованы для разработки собственной системы управления проектами и задачами.

Также были рассмотрены типы программного обеспечения, а также сделан вывод, что наиболее подходящими для разработки систем управления проектами являются веб-приложения. Такие приложения могут использоваться на различных устройствах, благодаря чему могут предоставлять быстрый доступ к системе.

Мир стремительно меняется, и компаниям необходимо подстраиваться под новые условия рынка. Системы управления проектами предоставляют компаниям такую возможность, поэтому их создание и использование является очень актуальной темой в наше время.

Список литературы

1. Ковалевская Н.А. Управление ресурсами на предприятии в контексте задач управления проектами. // Интеллектуальный и ресурсный потенциалы регионов: активизация и повышение эффективности использования. –2019. –C. 55-61.
2. Цыплов Е.А., Новиков В.А., Хайитов Х.О. Применение информационных систем управления проектами в повышении эффективности системы управления качеством. // Форум молодых ученых. –2019. –№10(38). –C. 556-560.
3. Гнедых В.А. Что такое корпоративная система управления проектами и зачем она нужна компании. // Финансовый университет при Правительстве РФ. –2015. –№4-4(48). –С. 57-61.
4. Методологии управления проектами: 12 популярных подходов [Электронный ресурс]. // Asana. URL : https://asana.com/ru/resources/project-management-methodologies (дата обращения: 28.04.23).
5. Лозгачева Т.М., Табекина О.А., Федотова О.В. Agile и научная организация труда: практика применения гибких методов в России. //Ученые записки крымского федерального университета. –2019. –№2. –C. 48-59.
6. Коул Р. Блистательный Agile. Гибкое управление проектами с помощью Agile, Scrum и Kanban. // Санкт-Петербург: Питер. –2019. –304 с.
7. Бахиркин М.В., Лукин В.Н. Как собрать кубик рубика: Waterfall или Agile в современных ИТ-проектах. // Материалы XXII международной конференции по вычислительной механике и современным прикладным программным системам. –2021. –C. 636-638.
8. Жук А.И. Управление задачами в Agile и Waterfall. // Студенческий. –2019. –№1-1(45) –C. 35-37.
9. Котляр Е.В. Система управления проектами Канбан. // Бизнес-образование в экономике знаний. –2020. –№ 1. –С. 57-60.
10. Плюснина Е.В., Орлова С.Н., Плеханов С.В. Практическое применение системы Kanban на разнопрофильных предприятиях. // Проблемы развития современного общества. –2022. –C. 134-138.
11. Scrum vs Waterfall vs Agile vs Lean vs Kanban [Электронный ресурс] // Visual Paradigm. URL : https://www.visual-paradigm.com/scrum/scrum-vs-waterfall-vs-agile-vs-lean-vs-kanban/ (дата обращения: 28.04.23).
12. Jira | Программное обеспечение для отслеживания задач и проектов [Электронный ресурс]. // Atlassian. URL : https://www.atlassian.com/ru/software/jira (дата обращения: 28.04.23).
13. Управление проектами команды с любого устройства | Trello [Электронный ресурс]. // Trello. URL : https://trello.com (дата обращения: 28.04.23).
14. Управляйте работой, проектами и задачами своего коллектива в сети [Электронный ресурс]. // Asana. URL : https://asana.com (дата обращения: 28.04.23).