**Слайд 2**

Повсеместное распространение Интернета изменило взгляд на информационную поддержку научно-исследовательской деятельности. Всё больше и больше научно-исследовательских лабораторий создают свои страницы, сайты, порталы. В настоящее время это не столь даже престижно, сколько удобно для всех, ведь наличие своего представительства в интернете – это отличная возможность для взаимодействия с единомышленниками и людьми, заинтересованными в продвижении научной деятельности. В режиме on-line может вестись общение с пользователями, обработка сообщений из формы обратной связи, публикация новостей лаборатории, сведений об имеющихся технических проектах, научных публикациях, сотрудниках лаборатории, а также может предоставляться та или иная сопутствующая информация, документы, описания, контакты. Стоит отметить, что на основании имеющихся в базе данных лаборатории сведений, при необходимости, можно собирать любые статистические данные, помогающие проанализировать деятельность лаборатории.

**Слайд 3**

Современные методы построения сайтов представляют из себя четыре основных направления разработки:

1. использование онлайн-конструкторов, таких как Wix, uCoz, Jimbo и др.;
2. использование бесплатных систем управления содержимым (CMS);
3. самостоятельная разработка;
4. использование фреймворка.

Каждый из этих подходов обладает своими достоинствами и недостатками.

**Конструкторы**

Онлайн-конструктор сайтов является желанным инструментом для веб-энтузиастов и предпринимателей, помогая им с развитием малых и средних веб-проектов. Конструктор позволяет в кратчайшие сроки разработать любой тип сайта (визитки, интернет-магазины и т.д.), без необходимости глубокого изучения языков программирования, инструментов и утилит веб-разработки.

Несмотря на кажущиеся простоту и удобство использования конструктора веб-сайтов, данный метод разработки имеет существенные недостатки:

1. нет доступа к исходному коду движка;
2. ограниченный функционал, определенные сложности с внедрением внешних плагинов и скриптов;
3. плохая поисковая оптимизация, затрудняющая продвижение сайта в сети;
4. наличие сторонней рекламы, которую владельцы конструктора вывешивают на видном месте;
5. нет возможности перенести сайт на другой хостинг.

Перечисленные недостатки делают использование конструктора для создания веб-ориентированной системы для научно-исследовательской лаборатории неприемлемым.

**CMS**

Подавляющее большинство сайтов на сегодняшний день разработаны при помощи той или иной системы управления. Благодаря наличию такой системы, владельцы сайта могут наполнять его любой информацией, публиковать новые страницы и новости, принимать платежи и заявки, не имея при этом каких-либо навыков в области веб-программирования.

Многие движки также поддерживают систему модулей. Это означает, что вы можете расширить функционал системы за счет подключения дополнительных модулей (плагинов), например, плагин чата или плагин обратной связи.

Подход с применением готовой CMS, по сравнению с использованием конструкторов, позволяет создать достаточно качественный и многофункциональный проект. В то же время, избыточный функционал любой CMS далеко не лучшим образом сказывается на производительности системы. А тот факт, что исходные коды многих популярных CMS выложены в общий доступ, ставит под угрозу взлома все сайты, построенные на их основе.

**Самостоятельная разработка**

Написание самописного движка для системы является нелегкой задачей для любого разработчика. Однако, любой интернет-ресурс, разработанный с «нуля», позволит реализовать любые пожелания и индивидуальные потребности разработчика.

При использовании CMS не всегда получится найти сторонние плагины, целиком и полностью удовлетворяющие запросам пользователя. При попытке написания своего модуля для внедрения в существующую систему управления содержимым, могут возникнуть определенные трудности, связанные с незнанием внутреннего устройства CMS и принципов взаимодействия её компонент. При самостоятельной же разработке программист сам определяет архитектуру разрабатываемой системы и может добиться повышения производительности системы за счет устранения избыточных функций и связей.

Несмотря на то, что данный подход позволяет достичь более качественной разработки веб-приложения, он имеет главный недостаток: наличие больших затрат времени, а также трудоемкости работы. Без должных навыков в программировании и создании веб-сайтов велика вероятность получить на выходе «сырой» продукт, который будет уступать существующим техническим решениям по уровню качества и проработки.

**Фреймворки**

Фреймворк можно считать заготовленным шаблоном (структурой), управляющим большинством повторяющихся и общих для всех веб-сайтов особенностей.

Задача фреймворка состоит в том, чтобы позволить дизайнерам и разработчикам сфокусироваться на создании уникальных функций для своих проектов, а не изобретать велосипед путем написания одинаковых вещей, общих для большого числа веб-сайтов и веб-приложений.

Таким образом, делая сайт на базе фреймворка, мы получаем доступ к огромному количеству готовых функций. Все основные вещи – от доступа к базе данных до отдачи готовой страницы браузеру – уже написаны создателями фреймворка.

У каждого из них есть свои сильные и слабые стороны. Нельзя назвать объективно лучший фреймворк, который бы подошёл для любой ситуации и решал бы все проблемы.

**Вывод:**

Таким образом, реализация нашей веб-ориентированной системы с применением функционала фреймворка является наиболее оправданным решением. Во-первых, в отличие от самописных решений, это позволит добиться простоты сопровождаемости проекта; во-вторых, решения на фреймворках работают значительно быстрее и способны выдерживать большую нагрузку, чем CMS и самописные системы, а также превосходят их по уровню безопасности.

**Слайд 4**

Практичность языка PHP для ведения веб-разработок обусловлена его следующими важными характеристиками:

1. «движок» PHP устроен таким образом, что позволяет обрабатывать сценарии с достаточно высокой скоростью;
2. поддержка широкого круга баз данных;
3. поддержка большого количества аппаратных платформ и операционных систем, включая Linux, Windows, Mac OS X и многие другие;
4. поддержка большинства современных веб-серверов (Apache, IIS и др.);
5. гибкость по отношению к потребностям разработчика благодарю обилию различных библиотек и расширений

Диаграмма, представленная выше, доказывает, что язык PHP является наиболее предпочтительным в среде веб-разработки. Такую популярность он заслужил во многом благодаря легкости в его освоении на различных этапах и рассмотренным выше особенностям.

Зачастую, выбор определенного фреймворка зависит от его популярности и простоты в использовании, а также от личных предпочтений разработчика. Любой фреймворк только предлагает достаточно низкоуровневые инструменты для разработки веб-приложений – большинство функционала разработчику придется реализовывать самостоятельно.

Для достижения поставленной в работе цели был выбран PHP-фреймворк Yii. Yii – это высокоэффективный фреймворк общего назначения, который используется для разработки веб-приложений любого масштаба. Как и большинство других популярных PHP-фреймворков, Yii использует широко применяемый в веб-программировании шаблон проектирования Model-View-Controller (MVC). Превосходство Yii перед другими фреймворками заключается в его эффективности, широких функциональных возможностях и качественной документации (включая обширное русскоязычное сообщество). Yii изначально был спроектирован таким образом, чтобы максимально соответствовать всем требованиям при разработке современных веб-приложений.

**Слайд 5-6**

Один из основных принципов современной веб-системы состоит в том, что практически все её данные хранятся в базе данных, а пользователям предоставляется удобный интерфейс работы с системой. Администратору же системы для обновления не требуется вручную редактировать файлы на сервере — для этого создаются интерактивные веб-формы.

В дополнение к перечисленным преимуществам стоит также добавить, что данная СУБД обладает наилучшей скоростью обработки данных объемом до 500 тыс. записей

Это базисные таблицы, необходимые для работы сайта

Здесь стоит отметить, что СУБД MySQL гибка в этом отношении. Она позволяет менять структуру таблиц даже на развёрнутых, рабочих проектах.

**Слайд 7**

Каждый раз, когда пользователь заходит в систему, входной скрипт index.php загружает файл конфигурации web.php, после чего создает экземпляр приложения для дальнейшей обработки запроса. Приложение определяет маршрут, на который необходимо перевести запрос, при помощи встроенного в приложение компонента «запрос». Далее для выполнения запроса приложение создает экземпляр контроллера, который, в свою очередь, создает действие и накладывает на него необходимые фильтры (такие как фильтр доступа, фильтр по типу запроса и т.д.). После того, как все фильтры будут успешно пройдены, действие контроллера загружает модель данных из базы данных и генерирует представление, заполняя его данными из модели. Сгенерированное представление передается как компонент приложения «ответ». Компонент «ответ» отправляет результат работы приложения в браузер пользователя.