Существует очень много форумных CMS. Среди них есть как самостоятельные, так и в виде плагинов к CMS. Использование последних требует предварительной установки CMS. Рассмотрим некоторые CMS(показываем слайды, на каждую cms свой слайд)

* Плагин для WordPress – Simpleforum

WordPress — система управления содержимым сайта (CMS) с открытым исходным кодом, распространяемая под GNU GPL. Написана на PHP, в качестве базы данных использует MySQL. Сфера применения — от блогов до достаточно сложных новостных ресурсов и даже интернет-магазинов. Встроенная система «тем» и «плагинов» вместе с удачной архитектурой позволяет конструировать практически любые проекты.

* Simple Machine Forum (SMF)

Самостоятельная бесплатная форумная CMS, написанный на PHP с использованием базы данных MySQL. Одна из самых популярных. Так же очень функциональна – имеется множество настроек в панели администрирования, есть механизмы разделения прав по группам пользователей и тд.

* phpBB

phpBB (PHP Bulletin Board) —бесплатный веб-форум со свободным исходным кодом, разработанный на скриптовом языке PHP, поддерживающий различные СУБД, включая MySQL, PostgreSQL, MS SQL Server, MS Access, SQLite, а также Oracle (при наличии необходимых изменений Является самостоятельной CMS.

Все перечисленные системы обладают недостатками, такие как

* перегруженность интерфейса
* отсутствие русского языка в интерфейсе
* низкая производительность из-за сложных запросов к базе данных

Наша задача заключалась в том, чтобы разработать самостоятельную форумную CMS, в которой будут устранены данные недостатки, а так же она будет удовлетворять следующим требованиям и иметь следующие функции(на слайде списком)(тут можно упорядочить)

* форум состоит из категорий, которые включают в себя темы (топики), в которых идет обсуждение
* возможность закрытия темы для обсуждения
* создание темы
* создание сообщения
* удаление сообщения участником форума, имеющим такое право
* создание скрытых категорий, которые видны ограниченному количеству групп пользователей
* наличие панели администратора, через которую будет производится управление форумом
* страница регистрации новых пользователей
* модуль авторизации пользователей в системе
* при передаче конфиденциальных сведений от клиента к серверу, они должны быть зашифрованы
* защита от SQL инъекций
* наличие панели навигации по сайту
* масштабируемый механизм разделения прав
* CMS должна иметь определенную архитектуру, чтобы другие разработчики могли в ней разобраться

(слайд перечисление всех средств)

Весь использованный инструментарий делится на несколько типов

* Веб серверы
* Среды разработки
* Средства отладки программного кода
* Системы контроля версий программного кода
* Вспомогательные средства

(слайд скрин хампа)

Выбор сервера, исполняющего серверную часть приложения, обусловлен выбранными технологиями: PHP и MySQL. Для выполнения сценариев на языке PHP необходимо использовать сервер Apache. Для использования связки PHP+MySQL была использована сборка сервера XAMPP. Выбор обусловлен бесплатностью, отсутствием сложной настройки – готовность к использованию сразу после установки.

(слайд нетбинса)

Для написания кода была выбрана среда NetBeans IDE. Все файлы, из которых состоит веб сайт, данная среда объединяет в проект. Это удобно тем, что доступ ко всем файлам можно осуществить из одного места. Все редактируемые файлы разбиваются по вкладкам, что так же облегчает навигацию по файлам проекта. Среда поддерживает автодополнение, что существенно экономит время. Так же имеется подсветка синтаксиса.

(слайд XDebug)

Отладка серверной части происходила внутри выбранной среды разработки NetBeans средствами встроенного отладчика Xdebug. Встроенный отладчик имеет всю необходимую функциональность:

* точки останова;
* слежение за переменными;
* построчное выполнение;
* запуск произвольного скрипта.

(слайд отладки оперы и хрома)

При отладки клиентской части были использованы отладчик оперы и отладчик хрома. Данные отладчики имеют схожую функциональность, а именно:

* Просмотр DOM;
* Просмотр и редактирование CSS;
* Отладчик JavaScript;
* Просмотр HTTP запросов;

(слайд дерева джита)

Для организации совместной работы двух человек была использована система контроля версий GIt. Она позволяет объединять все изменения, внесенные разработчиками, а так же представляет историю изменений в виде дерева. В случае необходимости (такие как неизвестная ошибка), можно вернуться на предыдущее состояние проекта и начать работу с контрольной точки.

(вспосогательные средства – перечисление)

* Скайп
* Визио
* Браузеры хром и опера

Для функционирования любой CMS необходима база данных, в которой будет храниться весь контент. Проектирование базы данных было разделено на два этапа – логический и физический.

* Логический этап(слайд). На этапе логического проектирования необходимо выделить ключевые сущности и их свойства. При анализе требований к разрабатываемой CMS я пришел к этой структуре базы данных(показываю чертеж/плакат, перечисляю все таблицы и их свойсва) При этом, есть сущности, которые связаны между собой различными отношениями. Например, сущность «категории» связана со сущностью «темы» идентифицирующим отношением – ключевое поле «идентификатор категории» в сущности «тема» является внешним ключом к сущности «категория», где «идентификатор категории» является первичным ключом.
* Физический этап(слайд) На данном этапе база данных проектируется с учетом требований СУБД MySQL. (показываю чертеж/плакат).

После разработки архитектуры и проектирования бд, мы приступили к написанию модулей.

(слайд скрина страинцы реги)Модуль регистрации позволяет пользователям регистрироваться в системе. Необходимо ввести будущий логин, адрес электронной почты и пароль. Если какие либо данные будут некорректны, будет выведено предупреждение об этом. При отправке данных на сервер осуществляется их проверка на корректность не только на клиентской стороне при помощи javascript но и на сервере. Исключаются неверные адреса электронных ящиков и sql инъекции. Алгоритм работы валидации данных показан на слайде(слайд чертежа/плаката с алгоритмом)

(слайд со скрином редактируемых данных – пользователь редактирует свои данные) Модуль пользователя позволяет просматривать и редактировать данные. Здесь можно заполнить те поля, которые не заполнялись при регистрации. Доступ к редактированию данных есть только у их хозяина. Данных других пользователей можно только просматривать. (слайд – только просмотр данных) Так же пользователь может указать свою фотографию в профиле, написав ссылку на изображение. Все изображения необходимо загружать на сервис глобально распознаваемых аватаров Gravatar.com

(слайд со списком категорий) Сам форум состоит из категорий, которые содержат темы (слайд со списком тем). Внутри темы – сообщение пользователей. Пользователь может оставлять сообщение, если тема не закрыта. Закрыть тему для обсуждений может только уполномоченный пользователь. Так же уполномоченные пользователи могут удалять сообщения и темы (все это на слайд).

(слайд Димона – защита модулей)

(слайд с разделением прав) Для обеспечения гибкости предоставления контента, необходимо реализовать систему разделения прав. Чтобы грамотно это сделать, необходимо рассматривать всю систему как хранилище контента, где к каждой единице доступ регулируется системой прав. Кроме контента, система содержит действия пользователей, которые тоже должны регулироваться системой прав.

(картинка с вики про мд5) Так как пересылать пароли в незашифрованном виде небезопасно, было принято решение перед отправкой использовать шифрование мд5. При помощи javascript введенный пользователем пароль преобразуется в md5-строку и отправляется на сервер. На сервере все пароли хранятся тоже в зашифрованном виде. Если данные о пароле из бд и те, что пришли от клиента совпали – пароль был введен верно.

(заключение)