**Разработка интерфейса**

Для вывода данных пользователю используется язык гипертекстовой разметки HTML5 (различные его элементы, в частности аудио-проигрыватель) и CSS3 для графической составляющей интерфейса (цвета, шрифты и т.д.).

**JS + AJAX**

Интерактивность клиентской части придается за счет языка JavaScript и технологии AJAX.

JS выполняется на стороне клиента, и для этого не нужно взаимодействовать с сервером.

Технология AJAX позволяет асинхронно взаимодействовать с сервером.

Например, рассмотрим добавление слов в словарь.

Без AJAX необходимо было бы каждый раз перезагружать страницу, загружать много лишних данных. Так же каждый раз придется крутить с начала страницы до нужного места.

С AJAX слова добавляются асинхронно, и перезагрузка не требуется.

Еще одним примером является тестирование (например, написание слова).

При загрузке страницы, в клиент передается правильный ответ. Затем, когда пользователь отвечает в тесте, функция JS проверяет правильность и выводит информацию об этом пользователю, а AJAX отправляет запрос серверу на запись этих данных в БД. Все это происходит быстро и без перезагрузки страницы.

То есть JS и AJAX очень полезен как серверу, так и пользователю:

* На сервер приходится меньше нагрузки, так как JS выполняется на стороне клиента и благодаря AJAX передается меньший объем информации
* Клиентской части передается меньше лишней информации
* Страница лишний раз не обновляется и это выглядит удобнее и приятнее

**MVC**

В проекте реализован шаблон MVC (Model-View-Controller).

Поэтому вся программа разбита на несколько частей:

* Модель – база данных и работа с данными
* Вид – отображение данных в браузере пользователя
* Контроллер – контроль взаимодействие компонентов системы и взаимодействие с пользователем

Таким образом, пользователь делает какое-то действие в системе. Вид отправляет запрос Контроллеру. Контроллер распознает что необходимо сделать. Отправляет запрос в Модель. Если необходимо считать или записать какие-то данные в БД. Затем делает запрос в Вид на отображение необходимой информацию пользователю.

Благодаря шаблону MVC программу легче:

* Разрабатывать
* Тестировать и отлаживать
* Поддерживать и развивать

С MVC код программы становится более структурированным и понятным.

**Разработка БД**

В качестве СУБД была выбрана MySQL, так как ее легче всего интегрировать с веб-сайтом. А так же в PHP уже есть встроенные функции для работы с такой БД (подключение, выбор БД, отправка и обработка запроса и т.д.).

БД была нормализована по 3 нормальной форме.

**Администрирование**

Преподаватели кафедры иностранных языков в дальнейшем смогут сами поддерживать и расширять систему новыми материалами, используя административную панель.

**Развитие системы**

При желании данную программу можно без проблем дополнять новыми полезными функциями:

* Разработка сетевого приложения и взаимодействие между пользователями
* Разработка мобильного приложения
* Добавление языков интерфейса