**ОРГАНАЙЗЕР УЧЕБНОГО ПРОЦЕССА УНИВЕРСИТЕТА**

Я выбрал именно этот проект, так как уже имею некоторый опыт разработки аналогичных систем и web-сайтов в целом.

***[go]***

*Цель создания и преимущества.*

Целью создания данного проекта является:

* Экономия времени и сил сотрудников университета и студентов;
* Предоставление студентам возможности удаленного доступа к различной персональной информации и данных о преподавателях.

Я полагаю, что наибольшая польза от данного проекта будет именно для студентов. Так как я в большей степени рассматриваю эту систему со стороны студента (а всех тонкостей деятельности сотрудников не знаю).

***[go]***

Как студент я сталкивался с множеством проблем во время обучения. И некоторые из них можно будет решить с помощью разрабатываемого мною приложения. Самое главное, что студентам не нужно будет ездить в университет, бегать по нему и искать нужную информацию, искать работников и отвлекать их.

Студенты легко смогут узнать такую информацию:

* Личные данные и финансовые платежи;
* Свои текущие и финальные оценки;
* Свою посещаемость;
* Задания, требования, вопросы и сроки семинаров, практических и лабораторных работ;
* Расписание своих занятий и других событий (консультаций, отработок и т.д.);
* Расписание занятий, консультаций и отпусков преподавателей. А также возможность записать на консультацию.

***[go]***

Для сотрудников университета данный проект также предоставляет множество преимуществ:

* Легко вести журналы оценок и посещаемости;
* Автоматически генерировать различные статистики и отчеты;
* Хранить и работать с финансовыми операциями;
* Удаленно просматривать любую необходимую информацию.

***[go]***

*Критерии и методы разработки.*

Проект разрабатывается в виде web-сайта с использованием языков HTML, CSS (для графического интерфейса), JS, а так же AJAX и JQuery (для клиентского приложения и для интерактивности), PHP (для серверной части) и SQL (для обращения к БД). Так же проект основывается на парадигме ООП (это касается в основном PHP) и шаблоне MVC (весь код разбит на части: модель, контролер и представление).

Такой выбор позволяет легко реализовывать клиент-серверное взаимодействие и пользовательский интерфейс.

Так же благодаря этому пользователи смогут воспользоваться программой, имея лишь компьютер (или ноутбук, планшет, телефон) и доступ в интернет.

***[go]***

Отдельно стоит выделить требование к БД и серверу

База данных разработана на MySQL-сервере. Это обосновано относительной легкостью её интеграции с web-сайтом.

Основными требованиями к БД являются безошибочное выполнение запросов и достаточно высокая скорость их выполнения (так как приложение является интерактивным и пользователи не захотят им пользоваться, если оно будет работать долго либо с ошибками).

Требованиями к серверу БД являются:

* Достаточно большой объем памяти;
* Круглосуточная и безотказная работа даже при большом количестве запросов.

***[go]***

Основными критериями разработки являются простота, надежность и удобство использования программы.

Все должно быть интуитивно понятно, и даже начинающий пользователь должен суметь найти и разобраться с тем, что и как работает.

В проекте не используются супер-современные функции и навороты. Так как для данного проекта это лишнее. Гораздо более полезным будет то, что его можно будет не сложно открыть даже с плохим интернет соединением. Единовременно не загружается слишком много данных, используется минимум картинок. Даже если загружается большой список, то используется пагинация и подгружается только часть списка.

Для графического интерфейса выбрана сине-бело-серая цветовая гамма (можно подбирать оттенки). Она является нейтральной и не напрягает зрение пользователя даже при длительном использовании. Лично я работаю с такими сайтами по несколько часов в день и не испытываю никакого дискомфорта.

***[go]***

*Разделы и функции.*

В проекте присутствуют следующие разделы:

* Структура университета – информация о факультетах, кафедрах, специальностях и группах;
* Настройки – различные системные настройки, пользовательские настройки, списки ролей и должностей, финансовые (платежи) и образовательные (расписание пар, система оценок) настройки;
* Пользователи – личная информация о пользователях и данные для авторизации;
* Студенты – личная информация о студентах, их предыдущем образовании, текущих и прошлых группах, а так же данные для авторизации;
* Финансы – хранение финансовых операций студентов;
* Обучение – хранение оценок, посещаемости, заданий и т.д.

***[go]***

*Управление проектом и безопасность.*

Для доступа к проекту пользователь обязан авторизироваться в систему. Для авторизации используется действующий Email пользователя и личный пароль. Пароль хранится в БД в хэшированном по функции md5 виде.

В проекте внедрена система прав пользователей и студентов, поэтому управлять им довольно просто. Система прав не очень большая, в основном по одному или несколько прав на раздел. У каждого права есть 3 варианта значения:

* Нет – право доступа отсутствует;
* Просмотр – можно только просматривать данные, но модификация запрещена;
* Просмотр и Изменения – разрешены просмотр и модификация данных.

Таким образом все пользователи будут иметь доступ лишь к той информации, которая им нужна.

А студенты смогут лишь просматривать личную информация и информацию об университете и преподавателях.

Неавторизированные пользователи или пользователи без нужных прав не будут видеть лишние кнопки и функции, которые им не доступны, и не смогут зайти на недопустимые им страницы даже через url.

***[go]***

*Поддержка и модернизация.*

Поддержка проекта (дополнение новыми функциями и багфиксы) может производиться программистами, которые разбираются в web-программировании и имеют базовые навыки в системном администрировании, возможно, даже студентами. Проект является относительно не сложным, понятным для изучения и легким для модификации.

Система может постоянно развиваться, улучшаться и подстраиваться под потребности пользователей. Например, проект можно дополнить такими функциями:

* Разработка мобильного приложения или мобильной версии сайта для большей удобности использования;
* Автоматическое выставление посещаемости студентов (студенты смогут сами отмечать, что они присутствуют на занятии);
* Проведение финансовых операций с помощью API банков и других платежных систем;
* Напоминания в виде Email и SMS;
* Электронная подпись документов с использованием уже готовых и проверенных API (например, RightSignature);
* Университетский чат между студентами и преподавателями;
* Формирование различных статистик, отчетов и ведомостей.