

РАЗРАБОТКА ПЛАТФОРМЫ ДЛЯ КООРДИНАЦИИ НАУЧНЫХ КОНФЕРЕНЦИЙ

DEVELOPMENT OF A PLATFORM FOR THE COORDINATION OF SCIENTIFIC CONFERENCES

A. Pimanov

Summary: An important aspect for the implementation of any project is the choice of optimal solutions, taking into account its features, deadlines and budget. Therefore, the right choice of tools, implementation methods and functional features is the main task to achieve the successful completion of the project.

This article describes the possible implementation of the terms of reference for the development of a platform for the coordination of scientific conferences. The tools and methods that can be used to design such platforms are also presented, a variant of a possible architecture is given and the programming languages used, data storage methods and interfaces are considered.

Keywords: website, PHP, CSS, JavaScript, JSON, AJAX.

Согласно техническому заданию, разрабатываемый сайт должен являться агрегатором научных конференций, предоставлять посетителям возможность подавать заявки на участие в интересующих их научных конференциях, а также изменять или отменять поданные заявки. К перечисленному функционалу, прилагается и сопутствующий, который будет описан ниже.

Хороший дизайн помогает улучшить восприятие информации, повышает уровень доверия к сайту и его авторам, а также улучшает пользовательский опыт. Для поставленной задачи, дизайн сайта можно было сверстать с помощью многочисленных конструкторов, однако можно пойти и более сложным путём — воссоздать внешний вид сайта, используя непосредственно язык гипертекстовой разметки HTML, язык стилей CSS и язык программирования JavaScript. Такой подход является более правильным для практикующихся разработчиков.

Элементы управления — это элементы пользовательского интерфейса, с которыми пользователи взаимодействуют. Элементы управления можно спроектировать самостоятельно или использовать уже имеющиеся. Найти их достаточно просто. Существует множество ресурсов, где приводится не только реализация таких элементов, но и их исходный код. При включении подобных элементов в свой проект, велика необходимость добавления в код проекта вспомогательных инструментов. В роли таких инструментов могут выступать различные библиотеки и фреймворки. Например, для реализации

Пиманов Андрей Евгеньевич
Технический институт (филиал)
федерального государственного автономного
образовательного учреждения высшего образования
«Северо-Восточный федеральный университет
имени М.К. Аммосова», г. Нерюнгри
epimanov15@gmail.com

Аннотация. Важным аспектом для реализации любого проекта является выбор оптимальных решений, с учётом его особенностей, сроков и бюджета. Поэтому, правильный выбор инструментов, способов реализации и функциональных особенностей, является главной задачей для достижения успешного завершения проекта.

В данном материале описана возможная реализация технического задания по разработке платформы для координации научных конференций. Так же представлены инструменты и способы, которые могут быть использованы для проектирования подобных платформ, приведён вариант возможной архитектуры и рассмотрены используемые языки программирования, методы хранения данных и интерфейсы.

Ключевые слова: сайт, PHP, CSS, JavaScript, JSON, AJAX.

данной платформы потребовалось включить Bootstrap и Tailwind CSS.

Bootstrap — это бесплатный набор инструментов, который содержит в себе готовые шаблоны HTML и CSS для различных компонентов веб-интерфейса, таких как типографика, веб-формы, кнопки, метки, блоки навигации и так далее.

Tailwind CSS — это фреймворк CSS, который отличается от других, таких как Bootstrap, тем, что не предоставляет готовых наборов классов для элементов. Вместо этого он создает уникальный список «утилитарных» классов CSS, которые можно применять для стилизации каждого элемента, путем комбинирования и наложения этих классов.

Данные инструменты оказались полезными в проекте и после того, как были добавлены элементы управления, которые требовали их использования. Главная их польза выражалась в заготовленных CSS классах, позволяющих не дублировать повторно стили в других файлах CSS.

С текущим набором языков, фреймворков и библиотек можно реализовать главную страницу сайта. Данная страница должна содержать шапку, блок расписания и блок с информацией о выбранной конференции. Шапка должна содержать логотип, меню поиска и набор кнопок. Для того чтобы придать веб-странице желаемый

мое поведение, необходимо применять язык программирования JavaScript. Данный язык также понадобится для взаимодействия с невидимой пользователю частью сайта, отвечающей за его логику и хранение данных. Существенно облегчить реализацию выполняемых языком задач поможет библиотека JQuery. Данная библиотека представляет из себя набор функций JavaScript, фокусирующихся на взаимодействии с JavaScript и HTML. Библиотека JQuery помогает легко получать доступ к любому элементу гипертекстовой разметки, обращаться к атрибутам и содержимому элементов и манипулировать ими. Главным её преимуществом является то, что она способна предоставить удобный API для работы с AJAX позволяющим в фоне производить обмен данными между браузером и веб-сервером. Иначе говоря, можно производить такой обмен данными, не прибегая к перезагрузке веб-страницы.

Связка JavaScript и JQuery нашла своё применение в программировании элементов пользовательского интерфейса. К ним относятся кнопки, выводящие дополнительную информацию или вызывающие формы ввода данных, предварительно очищающих их поля, а также прочие поля ввода информации.

Однако привнесённых возможностей недостаточно для функционирования платформы. Необходимо организовать работу с данными которыми браузер и веб-сервер будут обмениваться. Организовать эту работу со стороны веб-сервера поможет С-подобный скриптовый язык общего назначения, интенсивно применяемый для разработки веб-приложений — PHP. Существует множество языков программирования, которыми можно было бы заменить PHP для решения рассматриваемой задачи. Однако в отличии от любого другого претендента на роль PHP, его поддержка наиболее вероятно не будет реализована на хостинге. В настоящее время PHP поддерживается подавляющим большинством хостинг-провайдеров.

С помощью PHP для данного проекта можно реализовать свой API, который будет принимать POST запросы, обрабатывать их и впоследствии возвращать результаты их выполнения. API (Application Programming Interface) — это набор определенных правил, протоколов и инструментов, которые позволяют разным приложениям и сервисам взаимодействовать между собой и обмениваться данными. Чтобы различать запросы между собой, каждый входящий POST запрос можно комплектовать полем type, а сопутствующие данные вкладывать любым удобным образом.

Для хранения данных, необходимо развернуть СУБД (систему управления базами данных). Любой хостинг-провайдер, предоставляющий услуги хостинга по умолчанию, включает в тарифы развёрнутую СУБД. Поэтому

разработчику остаётся лишь приступить к проектированию базы данных. База данных рассматриваемого проекта должна содержать таблицу для хранения проводимых конференций. Её структура выглядит следующим образом:

Наименование	Тип данных	Описание
ID	int	Номер записи в таблице, полю задан первичный ключ.
name	varchar(500)	Название конференции.
city	varchar(60)	Город проведения конференции.
place	varchar(500)	Место или адрес проведения конференции.
start	int	Дата начала конференции в формате Unix time.
end	int	Дата окончания конференции в формате Unix time.
description	text	Описание конференций.
events	json или longtext	Список одобренных заявок на участие в конференции.
applications	json или longtext	Список поданных заявок на участие в конференции.
information_letter	json или longtext	Список информационных писем, содержащих информацию об изменениях в проведении мероприятия.

Главными полями для списков events и applications являются: название доклада; авторы; наименование вложения. Для списка information_letter этими полями являются заголовок письма и его содержимое. В дополнении к списку events включается поле time хранящее время в формате Unix time.

Ещё одним необходимым инструментом, которым обязательно необходимо овладеть для разработки подобных решений — SQL. SQL (Structured Query Language) — это язык программирования, который используется для управления реляционными базами данных. SQL позволяет создавать, изменять и управлять базами данных, включая хранение, извлечение и обновление данных. Без данного инструмента, осуществить эффективную работу с данными невозможно.

При загрузке главной страницы, JavaScript используя AJAX отправляется POST запрос для получения списка научных конференций. Запрос комплектуется полем type со значением get_conferences. Со стороны веб-сервера данные извлекаются из базы данных, сортируются, время из Unix time переводится в привычный формат для пользователя и в завершении все данные собираются в JSON и возвращаются обратно. Из возвращённого JSON формируются блоки с конференциями

в виде расписания. Для получения подробной информации о конференции, следует спроектировать ответ для запроса со значением поля `type` — `get_conference_detail`. Данный запрос, получая идентификатор конференции должен возвращать список участников, заявок (при наличии авторизации) и список информационных писем (вне зависимости от наличия авторизации). Реализация и назначение личного кабинета будет рассмотрены ниже. Все данные списки будут вложены в один возвращаемый JSON запрос после сортировки. Полученные от веб-сервера списки будут реализованы в соответствующих блоках в виде списков.

Рассматриваемая платформа не может обойтись без форм для сбора информации. Формы состоят из нескольких элементов, включая поля ввода текста, флажки, кнопки и так далее, которые позволяют пользователю заполнить информацию и отправить её на сервер. Первая форма, которую встретит пользователь на главной странице сайта, необходима для подачи заявки на участие в выбранной конференции. Наличие текстовых полей ввода в данной форме обязательно. При помощи AJAX все поля придут в виде POST запроса, к которым легко можно будет получить доступ через суперглобальный массив. Трудности заключаются в обработке полей для загрузки вложения, так как к ним выдвигается больше требований. Чтобы пользователь при отправке вложения, не подходящего по выставленным критериям, не ожидал результатов проверки файла от веб-сервера — необходимо реализовать аналогичную проверку со стороны пользователя средствами JavaScript и JQuery. Загружаемый файл согласно техническому заданию не должен по объёму превышать 4 Мб и обязательно должен носить расширение pdf. Для отправки формы общий вид JavaScript кода будет выглядеть так:

Листинг 1

```
$(«наименование формы»).submit(function (e) {
    e.preventDefault();
    if ($('#наименование поля ввода для загрузки файлов').
        get(0).files.length !== 0) {
        var fileInput = $('#наименование поля ввода для загрузки
        файлов');
        var fileSize = fileInput[0].files[0].size;
        if (fileSize > 4 * 1024 * 1024) {
            return;
        }
        var form_data = $(this).serialize();
        $.ajax({
            url: 'адрес отправки данных',
            type: 'POST',
            data: new FormData($('#наименование формы')[0]),
            success: function (result) {}
        });
    }
});
```

Выполнение `e.preventDefault()` позволяет передать данные веб-серверу без перезагрузки страницы. После проверятся наличие вложения. Если оно было прикреплено проверяется его объём. В случае превышения файлом объёма 4 Мб, функция будет предварительно завершена. При этом функция ничего не вернёт. Если ранее перечисленные проверки не приведут к завершению скрипта, все данные с формы будут собраны в переменную `form_data` и с помощью AJAX отправлены на веб-сервер. В поле `success` заложена функция, которая выполняется после получения результата от веб-сервера. Для проверки на сервере полученного файла следует выполнить следующий PHP код:

Листинг 2

```
$filename = $_FILES['filename']['name'];
$extension = pathinfo($filename, PATHINFO_EXTENSION);
if (!in_array($extension, Config::ALLOWED_EXTENSIONS)
    or $_FILES['filename']['size'] > Config::MAX_FILE_SIZE
    or !move_uploaded_file($_FILES['filename']['tmp_name'],
    dirname(__FILE__, 2) . '/files' . strrchr($_FILES['filename']
    ['tmp_name'], '/') . '.pdf')) {
    exit("{\"result\": \"error_upload_file\"}");
}
```

Данный код проверит загруженный файл на соответствие расширению pdf и объёму в 4 Мб, также данный код переместит файл в заготовленную директорию к остальным файлам. Если файл не будет соответствовать критериями, выполнение кода прервётся и будет возвращена ошибка о загрузке файла. Полученную ошибку со стороны браузера можно вывести в виде диалогового окна, попросив пользователя проверить файл на соответствие критериям. В случае, когда претензий ни к файлу, ни к содержимому полей для текстового поля нет, необходимо запросить из базы данных список текущих заявок в формате JSON, вставить новую заявку и внести изменённый список в запись соответствующей конференции.

Большинство форм и их методов обработки данных со стороны веб-сервера устроены схожим образом с рассмотренной только что реализацией. Такими формами являются: добавление конференции; редактирование конференции; удаление конференции; добавление доклада; редактирование доклада; удаление доклада; одобрение заявки; редактирование заявки; удаление заявки; создание информационного письма; удаление информационного письма.

Перечисленные формы и их методы должны быть доступны только администраторам платформы. Чтобы скрыть и защитить их, необходимо реализовать возможности авторизации. Для решения данной задачи, в базу данных добавлена таблица со следующей структурой:

Наименование	Тип данных	Описание
login	varchar(100)	Логин администратора, полю задан первичный ключ.
password	varchar(100)	Пароль администратора.

На главную страницу добавлена форма авторизации с текстовыми полями для логина и пароля. В PHP скрипт внесена функция поиска администратора по приходящему логину и паролю. В каждый метод из ранее перечисленных, добавлена данная функция, а также спроектирован отдельный метод для формы авторизации содержащий только её. Если в результате выполнения метода найден соответствующий администратор, метод должен вернуть JSON запрос, содержащий информацию о своём успешном выполнении. На стороне пользователя должны быть сохранены логин и пароль как cookies. Cookies — это небольшие текстовые файлы, которые сайт сохраняет на компьютере пользователя. Они используются для хранения некоторой информации о пользователе и его действиях на сайте. Таким образом, можно запомнить логин и пароль, чтобы администратор не вводил его заново каждый раз. Однако теперь требуется при каждой отправке формы, дополнительно комплектовать набор данных логином и паролем взятых из cookies сайта.

На текущем этапе, пользователь может подать заявку на участие в конференции, но стоит предусмотреть возможность редактирования и удаления заявки. При разработке платформы было решено реализовать эту возможность следующим образом, описанным далее. Пользователь при подаче заявки обязательно должен указывать адрес своей электронной почты. Следовательно, чтобы подтвердить, что заявку хочет изменить или удалить именно тот, кто её подал, можно на указанный почтовый адрес отправить кодовое слово и просить пользователя ввести данное кодовое слово. Так как редактирование и удаление существующих заявок устроено достаточно сложно, было принято решение разместить данный функционал на отдельной странице. Данная страница встречает пользователя с формы подтверждения двумя текстовыми полями. Первое используется для ввода адреса электронной почты, а второе для кодового слова. Кодовым словом может быть что угодно и сгенерировать его можно любым удобным способом. Чтобы пользователю пришло кодовое слово, после того как он введёт адрес своей электронной почты, необходимо внедрить в проект PHPMailer. PHPMailer — это библиотека PHP, которая используется для отправки электронных писем из PHP-скриптов. Она предоставляет удобный интерфейс для работы с SMTP-серверами и поддерживает различные методы аутентификации, включая SMTP-аутентификацию, CRAM-MD5, LOGIN и PLAIN. Благодаря широкому набору способов аутентификации, произвести отправку письма можно почти

с любой электронной почты. Сгенерированное кодовое слово необходимо сохранить, чтобы проверить, что введёт пользователь во второе текстовое поле, с тем, что было отправлено на соответствующую почту. Для этого добавлена в базу данных таблица со следующей структурой:

Наименование	Тип данных	Описание
email	varchar(400)	Адрес электронной почты пользователя, полю задан первичный ключ.
code	varchar(6), длина будет зависеть от длины кодового слова	Кодовое слово, которое должен ввести пользователь.
time	int	Время создания кодового слова в формате Unix time. Используется чтобы наделить каждое кодовое слово «сроком жизни».

Если пользователь правильно ввёл кодовое слово, следует организовать для него временное пространство на подобии личного кабинета, в котором он сможет отредактировать или удалить заявки. Главная задача такого подхода — это обеспечить безопасность данных. Поэтому после прохождения авторизации логин и кодовое слово сохраняются как cookies. В данной связке кодовое слово будет выступать в роли пароля. Чтобы сохранить текущую сессию была создана ещё одна таблица в базе данных со следующей структурой:

Наименование	Тип данных	Описание
ID	int	Номер записи в таблице, полю задан первичный ключ.
email	varchar(400)	Адрес электронной почты пользователя.
code	varchar(6), длина будет зависеть от длины кодового слова	Кодовое слово.
time	int	Время создания сессии пользователя в формате Unix time. Используется чтобы наделить каждую сессию «сроком жизни».

Сессии следует удалять через определённое время, чтобы обезопасить данные. Для вышеописанных действий было выделено два метода: первый — отвечал за генерацию кодового слова, создание в таблице новой записи о попытке авторизации, удалении устаревших попыток авторизации и отправку письма; а второй — за проверку кодового слова и создание пользовательской сессии, посредством внесения в соответствующую

таблицу новой записи. В PHP код добавлена функция для проверки существования пользовательских сессий по адресу электронной почты и кодовому слову. Такую функцию содержат все методы, изменяющие данные в таблицах, описанные ниже, чтобы обеспечить безопасность данных.

После авторизации, пользователь должен получить список всех заявок, во всех конференциях в которых он участвует. Для этого можно изменить уже существующие методы или добавить новые. После извлечения из базы данных информации о всех заявках пользователя, следует вывести их в виде списка с соответствующими элементами управления для открытия форм редактирования и удаления заявки. Устройство этих форм, как и их методов может почти полностью совпадать с аналогичными им формами и методами для администраторов. Отличия заключаются в том, что их верификация должна производиться соответствующей функцией, опирающейся на данные сессий. А все заявки, в которые вносятся

изменения или удаляются, должны содержать один и тот же электронный адрес, на который был отправлен код подтверждения.

Теперь можно утверждать, что структура всего проекта описана в полной мере. Разработанный сайт полностью выполняет свои задачи, однако стоит учитывать возможности изменения концептов в проведении научных конференций. Поэтому подобные решения необходимо разрабатывать с учётом будущих изменений. Важно осознавать, что внедрение подобных решений с ориентацией на будущее может привести к усложнению устройства проекта. В связи с чем необходимо организовывать весь проект таким образом, чтобы обеспечить целостность данных, так как они попадают в группу риска и могут быть повреждены или утеряны. Данный материал содержит возможную методику проектирования подобных веб-сайтов, описывает инструменты и особенности, с которыми предстоит столкнуться веб-разработчику.

ЛИТЕРАТУРА

1. Котиева, Х.М. Язык разметки гипертекста HTML. Редакторы для HTML / Х. М. Котиева // . — 2020. — № 29–1(115). — С. 22–24. — EDN JCNSUO.
2. Официальный веб-сайт «php.net», раздел «Руководство по PHP» [Электронный ресурс]. URL: <https://www.php.net/manual/ru/index.php> (дата обращения: 26.04.2023).
3. Карпова И.П. Базы данных. — М.: Питер, 2013. С. 3–31.
4. Кузнецов М.В., Симдянов И.В. Самоучитель PHP 7. — СПб.: БХВ-Петербург, 2018. С. 143–151.
5. Джеймс Р. Грофф, Пол Н. Вайнберг, Эндрю Дж. Оппель. SQL. Полное руководство. — Вильямс, 2018. С. 95–148.
6. Радионов, С.В. Создание цветowych тем веб-сайта с помощью CSS и JavaScript / С.В. Радионов // Постулат. — 2019. — № 8(46). — С. 22. — EDN JODGBI.

© Пиманов Андрей Евгеньевич (epimanov15@gmail.com).

Журнал «Современная наука: актуальные проблемы теории и практики»