

DOI: 10.12731/2227-930X-2024-14-2-287
УДК 004.77



Научная статья | Системный анализ, управление и обработка информации

РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ ПРОВЕДЕНИЯ КОНФЕРЕНЦИИ «ПЕДАГОГИЧЕСКИЙ ПОТЕНЦИАЛ»

*О.В. Камозина, О.В. Охлупина,
К.С. Маганков, Н.П. Рябцев*

Актуальность темы исследования обусловлена необходимостью постоянного сотрудничества, обмена опытом среди преподавателей, учащихся различных образовательных учреждений. В то же время, требуется снижение затрат на организацию конференций, упрощение прохождения этапов подачи заявок, статей, оплаты оргвзноса, рассылки материалов и т.д. В связи с этим возникает необходимость внедрения цифровых технологий в проведение различных типов конференций.

Цель – разработка веб-приложения для проведения конференции.

Материалы и методы. Для достижения поставленной цели в статье определен технологический стек реализации проекта. Используются методы моделирования, анализа, синтеза.

Результаты. В работе описан процесс реализации технической и клиентской части веб-приложения для организации конференции. Выделены маршруты взаимодействия клиента с базой данных и отправки запросов на сервер приложения. Представлено создание базы данных веб-приложения, а также взаимодействие веб-приложения с сервером. Реализовано наполнение главной страницы. Определена форма подачи заявки на участие в конференции. Разграничены возможности участников и администраторов.

Область применения результатов. Веб-приложение было включено в проведение Всероссийской с международным участи-

ем научно-практической конференции «Педагогический потенциал», проходившей 27 ноября 2023 года в Брянском государственном инженерно-технологическом университете. Полученные результаты могут быть использованы для проведения конференций различного уровня.

Ключевые слова: конференция; веб-приложение; техническая часть; клиентская часть; модель; главная страница; участник; администратор

Для цитирования. Камозина О.В., Охлупина О.В., Маганков К.С., Рябцев Н.П. Разработка веб-приложения для проведения конференции «Педагогический потенциал» // *International Journal of Advanced Studies*. 2024. Т. 14, № 2. С. 123-144. DOI: 10.12731/2227-930X-2024-14-2-287

Original article | System Analysis, Management and Information Processing

DEVELOPMENT OF A WEB APPLICATION FOR THE CONFERENCE “PEDAGOGICAL POTENTIAL”

***O.V. Kamozina, O.V. Okhlupina,
K.S. Magankov, N.P. Ryabtsev***

The relevance of the research topic necessitated constant cooperation and exchange of experience between teachers and students of various educational institutions. At the same time, it is necessary to reduce the costs of organizing conferences, simplify the processes of submitting applications, articles, paying registration fees, distributing materials, etc. In this regard, there is a need to use digital technologies in holding various types of conferences.

Purpose – develop a web application for holding a conference.

Materials and methods. To achieve this goal, the article defines the technology stack for project implementation. Modeling, analysis, and synthesis methods were used.

Results. *The paper describes the process of implementing the technical and client parts of a web application for organizing a conference. The routes for interaction between the client and the database and sending requests to the application server are highlighted. The creation of a web application database is presented, as well as the interaction of the web application with the server. Filling of the main page has been implemented. The form for submitting applications for participation in the conference has been determined. The capabilities of participants and administrators are differentiated.*

Scope of application of the results. *The web application was included in the All-Russian Scientific and Practical Conference “Pedagogical Potential” with international participation, held on November 27, 2023 at the Bryansk State Engineering and Technology University. The results obtained can be used to conduct conferences at various levels.*

Keywords: *conference; web application; technical part; client part; model; home page; participant; administrator*

For citation. *Kamozina O.V., Okhlupina O.V., Magankov K.S., Ryabtsev N.P. Development of a Web Application for the Conference “Pedagogical Potential”. International Journal of Advanced Studies, 2024, vol. 14, no. 2, pp. 123-144. DOI: 10.12731/2227-930X-2024-14-2-287*

Введение

Информационное общество активно использует «экспансию нововведений», обеспечивающую конкурентные преимущества и определяющую стиль жизни общества [12, с. 4]. Неизбежным является факт, что переход от индустриального к информационному обществу влияет на организацию науки [4, с. 14]. Цифровизация трансформация, коснувшаяся всех областей жизни общества, затронула и сферу образовательной коммуникации [11, с. 77].

Образовательное и научное сообщество постепенно переходит к использованию возможностей «цифры» [3, с. 14; 8, с. 249] для повышения эффективности различных направлений своей деятельности. Внедрение инструментария информационных техно-

логий способствует обеспечению наиболее эффективных форм взаимодействия, в число которых входит не только обмен информацией о проводимых мероприятиях [10, с. 434], но и обеспечение среды для продуктивной работы по подготовке и размещению материалов [5, с. 35], оптимизацию её технической стороны, что важно как для организаторов, так и для участников.

Цель работы – разработка веб-приложения для проведения Всероссийской с международным участием научно-практической конференции «Педагогический потенциал».

Достижению цели работы способствовало решение следующих задач:

- проанализирован желаемый функциональный состав веб-приложения;
- создана модель приложения для проведения конференции;
- произведена программная реализация соответствующего приложения;
- выполнена апробация веб-приложения в реальных условиях;
- проанализированы результаты работы.

В рамках реализации проекта был выбран следующий *технологический стек*:

1. Построение Restful API сервиса – фреймворк Flask, библиотека Flask-Restx.

2. Хранение и управление данными – система MySQL, а также библиотека flask-sqlalchemy (для взаимодействия API интерфейсов с базой данных) [1, с. 192].

3. Построение пользовательского интерфейса веб-приложения – HTML5+CSS3+JS;

4. Построение фронтенд-части – фреймворк Flask, шаблонизатор Jinja, Bootstrap.

В рамках REST API сервиса были определены и реализованы необходимые маршруты (endpoints) для взаимодействия клиента с базой данных, а также отправки запросов на сам сервер приложения [13, с. 34]:

1. Users (управление пользователями системы);
2. Articles (управление статьями, сборниками);
3. Posts (управление информационными постами).

Реализация

Техническая часть

1. Разработка модели взаимодействия пользователя и системы

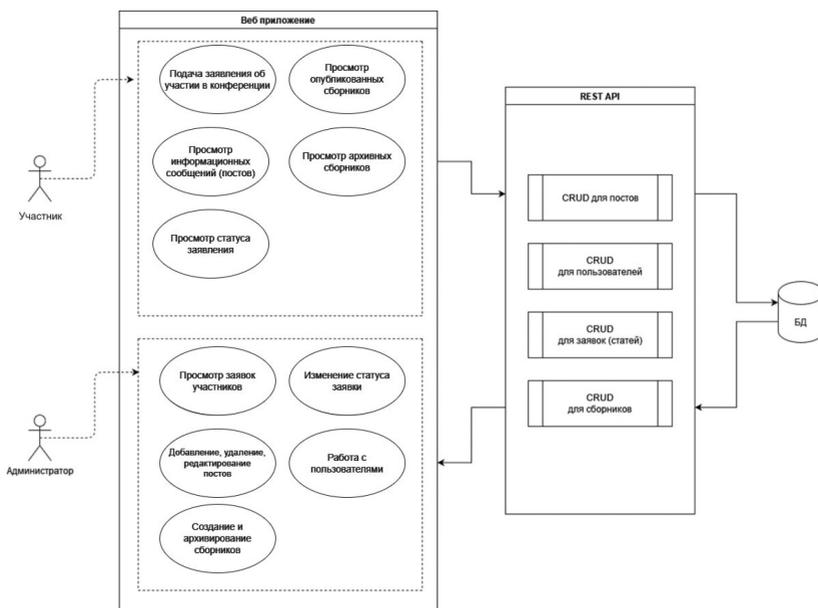


Рис. 1. UML Use-Case диаграмма

На данном этапе было организовано визуально-структурное представление приложения, а также связь его модулей между собой для формирования полноценной, цельной системы.

В модуле веб-приложения были обозначены подмодули, которые разграничивали логику приложения на два типа пользователей.

Первые являлись участниками, которые имели доступ к основным страницам приложения, таким как просмотр и подача заяв-

лений, просмотр постов, сборников, а также определение статуса для поданного заявления.

Вторые являлись непосредственно администраторами системы, у которых были права на добавление, редактирование, удаление информации, хранящейся в базе данных, изменение статуса для поданных заявлений участников и т.д.

Также был определен модуль REST API, который являлся прослойкой между веб-приложением и базой данных, организовывая связь, по которой веб-приложение могло свободно получать и отправлять информацию, без полного представления, как работает этот модуль [15, с. 73; 16].

2. Создание базы данных веб-приложения

Для хранения информации на сервере MySQL были определены следующие таблицы и сущности [6, с. 139; 7, с. 13]:

1. Таблица *users*, которая хранит информацию об администраторах системы, а также имеет связь с таблицей *posts*, постами, которые они публикуют.
2. Таблица *posts*, хранящая информацию об опубликованных администраторами постах, которые могут содержать сведения о предстоящих конференциях, выходах сборников и любые другие необходимые данные. Связь с таблицей *post_attachments* позволяет добавлять к тексту постов дополнительные приложения в виде файлов, изображений или других типов данных.
3. Таблица *articles*, которая хранит всю основную информацию о текущих статьях, включая информацию об участниках (*conference_authors*), о приложенных участниками файлах (*files*), таких как сама статья, квитанция об оплате участия в конференции, а также заявление.
4. Таблица *article_collections*, хранящая информацию об уже изданных сборниках, включая сборники, которых были заархивированы.

Графическая модель была перенесена в модель на основе классов для того, чтобы сервис мог корректно работать с базой

данных. С целью реализации этого были определены следующие классы [2, с. 207; 1, с. 273].

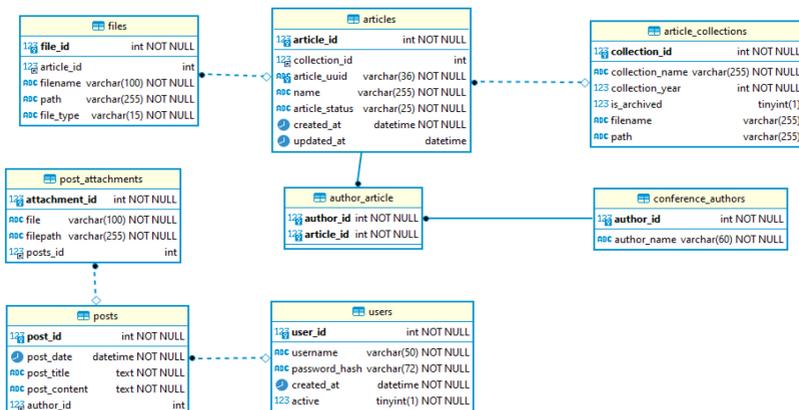


Рис. 2. Графическая модель базы данных

1. Классы для работы со статьями и авторами (articles):

– *Article*. Класс, содержащий информацию о заявках (статьях), в который входят следующие поля: *article_id* (первичный ключ), *collection_id* (идентификатор сборника, в который входит статья), *article_uuid* (уникальный номер заявки, по которому можно отследить ее статус), *name* (наименование статьи), *article_status* (статус заявки/статьи), *created_at* (дата создания), *updated_at* (дата последнего обновления).

– *ArticleCollection*. Класс, содержащий информацию о сборнике, в который входят следующие поля: *collection_id* (первичный ключ), *collection_name* (наименование сборника), *collection_year* (год выпуска/публикации сборника), *is_archived* (является ли сборник архивным); *filename*, *path* (наименование и путь к файлу сборника).

– *ConferenceAuthor*. Класс, содержащий информацию об участниках, в который входят следующие поля: *author_id* (первичный ключ), *author_name* (ФИО автора).

– *File*. Класс, содержащий информацию о приложенных файлах, таких как квитанция, заявление и сама статья, необходимых

для регистрации в конференции, в который входят следующие поля: `file_id` (первичный ключ), `article_id` (идентификатор статьи/заявки), `filename`, `path` (наименование и путь к файлу), `file_type` (тип файла: квитанция, заявление, статья).

2. *Классы для работы с постами (информационными сообщениями):*

– *Post*. Класс, содержащий информацию о постах, в который входят следующие поля: `post_id` (первичный ключ), `post_date` (дата публикации), `post_title` (заголовок поста), `post_content` (контент), `author_id` (идентификатор автора), `post_attachments` (приложенные файлы).

– *PostAttachment*. Класс, содержащий информацию о вложенных файлах, в который входят следующие поля: `attachment_id` (первичный ключ), `file`, `filepath` (наименование и путь к файлу), `post_id` (идентификатор поста).

3. *Класс для работы с пользователями:*

– *Users*. Класс, содержащий информацию о пользователях, в который входят следующие поля: `user_id` (первичный ключ), `username` (имя пользователя), `password_hash` (зашифрованная строка пароля), `created_at` (дата создания), `active` (активирован/нет), `posts` (выложенные пользователем посты).

Данные классы наследуются от `sqlalchemy database (db.model)`, что дает для них дополнительные параметры, позволяющие использовать больший функционал при взаимодействии с базой данных.

После создания модели классов также были созданы схемы (`marshmallow schemas`), позволяющие репрезентовать их массивы данных, проводить их обработку и преобразование как `json` строки/массивы.

Таким образом, мы создали как саму базу данных в среде `mySQL`, так и реализовали прослойку между сервером базы данных и приложением, чтобы иметь более гибкий контроль и простоту при работе с данными и их отправкой/хранением на сервере.

3. Взаимодействие веб-приложения с сервером (REST API)

1. Конечная точка *users*

Данное пространство имен *namespace* содержит в себе основные функции, которые реализуют взаимодействие с пользователем, например, создание нового пользователя, его изменение, удаление, а также методы аутентификации и выхода из системы.

Приведем пример кода, отвечающего за аутентификацию пользователя в системе:

```
data = ns.payload
user = User.query.filter(User.username == data['username']).first()

if user:
    if password_hash_compare(user.password_hash, data['password']):
        expires_delta = timedelta(days=15) if data.get('remember')
        else None

        access_token = create_access_token(identity=user.user_id, fresh=True,
expires_delta=expires_delta)
        return {
            'message': 'User login successful',
            'access_token': access_token
        }
    else:
        return messages.ErrorMessage.login_or_pass_error()
else:
    return messages.ErrorMessage.user_not_exist()
```

После того, как пользователь вводит логин и пароль в фронтенд части, это передается на бэкенд, где сверяется хэш введенного пароля и пароля, который хранится в зашифрованном виде в базе данных. Если логин или пароль неверны, то будет возвращаться ошибка, если данные совпадают с теми, которые записаны в базу данных, то будет создан специальный токен (JWT-token), который в дальнейшем станет использоваться для авторизации пользователя (администратора). [14, с. 9; 13, с. 121]

2. Конечная точка *articles*

Это *namespace* является ключевым, так как содержит основную логику приложения.

Основными функциями является:

- CRUD заявок, поданных участниками (статей);
- CRUD сборников, выпускаемых организацией;

- CRUD участников, которые участвуют в конференции;
- поиск статей, по уникальному номеру (UUID), присваиваемому при ее отправке;
- скачивание и просмотр файлов, приложенных к заявкам;
- изменение статусов заявок (отправлена, отклонена, принята);
- архивирование сборников;
- скачивание выпущенных сборников.

3. Конечная точка posts

Последнее из *namespace*, которое отвечает за минимальные функции выкладки информационных сообщений (постов).

Содержит в себе CRUD для постов, а также позволяет получить файл, прикрепленный к постам.

Клиентская часть

Разработка началась с определения HTML-страниц и их количества, которые необходимы для полноценного функционирования системы. Таким образом, было реализовано навигационное меню для перехода между данными страницами.

После этого возникла необходимость наполнить главную страницу контентом. Исходя из заданной задачи, появилась потребность в информативном блоке, который будет отображать в себе ключевую информацию. Именно поэтому было решено реализовать слайдер. Данный блок является относительно компактным, а также справляется с поставленной задачей, т.к. способен отображать достаточный объем информации в наиболее легко воспринимаемой форме – изображении. Имелась потребность в отображении нескольких слайдов, на этапе разработке туда были помещены заглушки с целью дальнейшей замены.

Далее началась разработка блока для вывода информационных сообщений – блога, основной задачей которого являлось отображение краткой ключевой информации о ходе конференции. Сам формат отображения информации представляет собой вывод заголовка, текста и вложенных файлов. Важно отметить, что воз-

возможность взаимодействия с блогом доступна только для администраторов веб-приложения.

Финальным этапом реализации главной страницы веб-приложения стало написание навигационного подвала, который представляет собой аналог навигационного меню с той разницей, что в меню показаны ссылки для перехода внутри приложения, а в подвале – ссылки на информационные ресурсы, расположенные вне приложения.

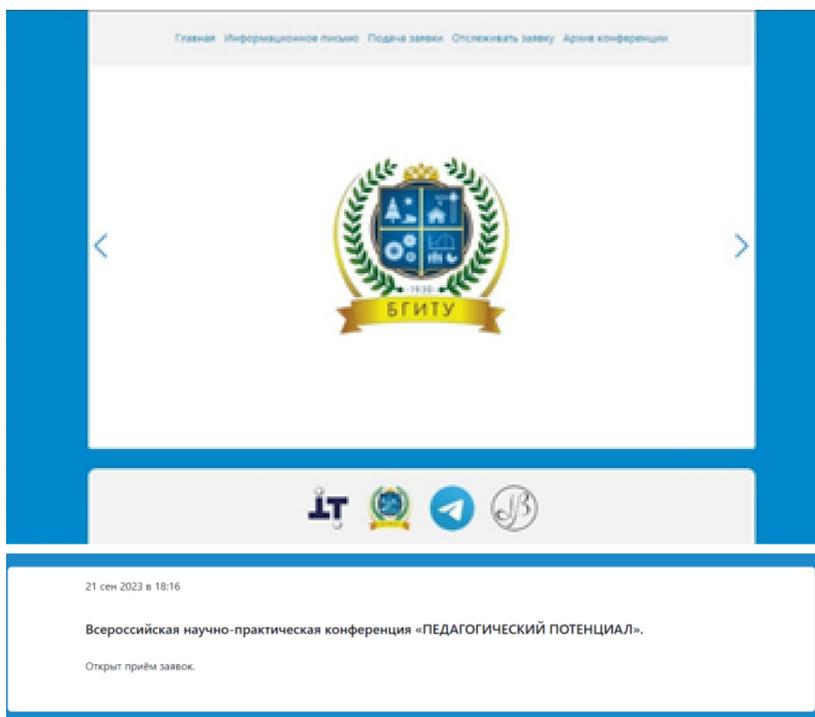


Рис. 3. Навигационное меню, слайдер, подвал, блог

Следующим этапом в реализации веб-приложения была разработка формы подачи заявки на участие в конференции. Данная часть являлась самой сложной с функциональной точки зрения.

Здесь помимо верстки необходимо было реализовать функционал взаимодействия клиентской части с серверной. А так как при подаче заявки задействуется информация сразу из нескольких таблиц, которые содержат сведения о статьях, участниках, сборниках и приложенных к заявке файлам, то это подразумевает написание нескольких запросов для получения определенных данных с целью их дальнейшей передачи в запрос на регистрацию заявки.

Для реализации верстки данной страницы потребовалось написание JS-скрипта, добавляющего карточку автора для подачи заявки. Это необходимо, так как в случае с научной публикацией у одной статьи может быть до 10 авторов.

Сам скрипт представляет собой код, который при нажатии на кнопку дорисовывает новые карточки с элементами формы. Для того, чтобы каждому полю ввода внутри карточки присваивалось уникальное имя, используются форматированные строки [16].

После реализации данного функционала была продолжена верстка основной формы, согласно утвержденной схеме данных – при регистрации помимо автора необходимо указывать название статьи, прикладывать файл об оплате, файл самой статьи и файл с заявкой на участие, где должны быть указаны все авторы, а также флажок, служащий для отображения получения согласия на обработку персональных данных – если участник не согласен, заявка подана не будет.

При написании запросов для обращения к серверу использовалась библиотека `requests`, с её помощью писались CRUD-запросы.

Так как при подаче заявки задействовались несколько таблиц, то сначала стоило понять, какие данные следует принимать от сервера и какие передавать.

Для успешной подачи заявки требовалось получать информацию о текущей конференции, для этого использовался GET-запрос с целью установления идентификатора актуальной конференции [17; 18].

После получения набора данных следовало извлечь из них конкретные значения, например, идентификатор или название конфе-

ренции, с целью дальнейшей передачи данных значений при формировании заявки.

Ф.И.О. автора
Автор 1

Ф.И.О. автора
Автор 2

Добавить автора

Название статьи

Квитанция об оплате
Выберите файл | Файл не выбран

Заявка на участие
Выберите файл | Файл не выбран

Статья
Выберите файл | Файл не выбран

Дать согласие на обработку персональных данных

Подать заявку

Рис. 4. Форма подачи заявки на участие в конференции

Кроме транспортировки информации с сервера, также передавалась информация с формы, заполняемой участником. Таким образом, информация об авторах собиралась в массив данных, после чего передавалась серверу. Аналогичный процесс происходил и с другими данными, извлекаемыми из формы.

Также важно отметить, что во избежание ошибок при подаче заявки, были добавлены условия для отработки тех или иных действий, например, если участник прикрепил файл неразрешенного формата.

После разработки и тестирования данного функционала, был реализован функционал для получения сведений о поданной заявке. Для этого после подачи заявки участника перебрасывает на новую страницу, где перед ним выводится уникальный номер его заявки, который служит для поиска и отслеживания.

В случае введения участником корректного номера заявки он получит информацию о статусе заявки. Если же данные некорректны или заявка была отклонена, после чего удалена администратором, участник получит сообщение о том, что ничего не найдено.

На момент реализации данного раздела информационное письмо уже было сформировано и размещено в соответствующем пункте меню.

Для конечной реализации CRUD-функционала заявок требовалось разграничение возможностей. Так участник может только подавать данные и производить поиск. Для администратора же требовалось возможность отклонения/принятия и удаления заявки.

Разработка администраторской части клиентской стороны веб-приложения началась с верстки форм авторизации и регистрации администратора, которые необходимы для успешной аутентификации с целью получения дополнительного функционала.

Когда формы были готовы, требовалось написать запросы для обращения на сервер с целью внесения данных об учётной записи или же, в случае с авторизацией, с целью прохождения процесса аутентификации, по завершении которой в систему сохраняется специальный ключ – JWT Bearer токен, служащий для предотвращения возможности несанкционированного взаимодействия с сервером [14, с. 44].

После реализации данного функционала необходимо было вывести список всех существующих участников и добавить возможность взаимодействия с их данными. Таким образом, была создана следующая страница, на которой администратор мог изменять/удалять/добавлять другие учётные записи.

После успешной авторизации администратор также попадал на главную страницу, но навигационное меню для него менялось в связи с функциональными возможностями.

Для завершения CRUD данных по заявкам на участие в научной конференции был реализован соответствующий интерфейс, в котором выводилась вся необходимая информация о заявке, а также имелись кнопки для принятия/отклонения/удаления заявки, возможность скачивания приложенных файлов, форма фильтрации заявок по их статусу.

Здесь также используется пагинация, как и на главной странице, которая служит для ограничения отображения количества записей на одной странице и их разбиения на страницы.

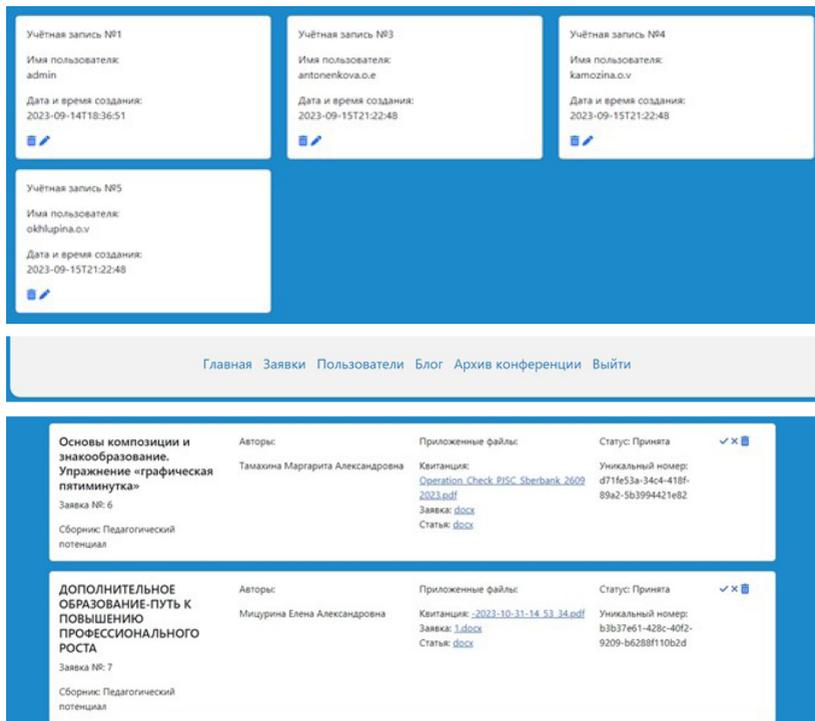


Рис. 5. Учётные записи, навигационное меню администраторов, форма отображения заявок участников

Далее оставалось реализовать функционал для архива конференции и блога сообщений. Процесс разработки данных функциональных компонентов был аналогичен предыдущим. Сначала необходимо было написать разметку, добавить стили, классы, после чего отследить значения из формы, передать их в запросе к серверу, или же наоборот, в случае с получением информации от сервера.

После окончательной реализации клиентской части была произведена проверка функционала системы с помощью набора тестовых данных. По её итогам внесены некоторые правки, повышающие безопасность приложения и определяющие поведение

системы в случае некорректных запросов, и снова запущена проверка, которая завершилась успешно. Это означало, что веб-приложение готово для размещения в пространстве Ethernet.

Заключение

Результатом работы явилось создание веб-приложения для проведения Всероссийской с международным участием научно-практической конференции «Педагогический потенциал» [9].

Для достижения цели работы был решён ряд задач: от анализа желаемого функционала приложения до создания его программной реализации и апробации.

В работе приведено описание реализации, содержащее ключевые моменты разработки модели взаимодействия пользователя и системы, создание базы данных веб-приложения и взаимодействие веб-приложения с сервером.

В ходе разработки были выбраны следующие инструменты: Flask, Flask-Restx, mySQL, flask-sqlalchemy.

Создана база данных в среде mySQL, реализована прослойка между сервером базы данных и приложением для обеспечения более гибкого контроля и простоты при работе с данными и их отправкой/хранением на сервере. Определен модуль REST API, организующий связь, по которой веб-приложение может свободно получать и отправлять информацию.

Разработка клиентской части включала работу с HTML-страницами, создание навигационного меню, разработку: информативного блока – слайдера; блога для отображения краткой ключевой информации о ходе конференции; навигационного подвала, содержащего ссылки на актуальные информационные ресурсы, системы авторизации, а также механизм избегания возможных ошибок; архива конференции и блога сообщений. Система была отлажена и успешно прошла итоговое тестирование.

Созданное веб-приложение позволило оптимизировать работу оргкомитета конференции и её участников.

Список литературы

1. Алан Бьюли. Изучаем SQL. [Пер. с англ.] М.: Символ-Плюс, 2017. 309 с.
2. Барри Поллард. HTTP/2 в действии. [Пер. с англ.] М.: ДМК-Пресс, 2021. 424 с.
3. Богданова И. Ф. Научные коммуникации в онлайн-пространстве // Наука и инновации. 2014. № 4 (134). С. 13-16.
4. Голованова Е.В. Прогностический потенциал теории постиндустриализма Д. Белла // Вестник славянских культур. 2013. № 4 (30). С. 11-17.
5. Гуськов А.Е., Васильков А.В. Средства поддержки проведения научных конференций: обзор и сравнение // Вестник Новосибирского государственного университета. Серия: Информационные технологии. 2010. Т. 8, № 4. С. 35-45.
6. Задорожный С.С., Фадеев Е.П. Объектно-ориентированное программирование на языке Python. М.: Физический факультет МГУ им. М.В. Ломоносова, 2022. 40 с.
7. Колисниченко Д.Н. PHP и MySQL. Разработка веб-приложений. Санкт-Петербург: Изд-во БХВ-Петербург, 2009. 607 с.
8. Мбого И.А., Прокудин Д.Е., Чугунов А.В. Разработка инструментов интеграции научной информации в пространстве разнородных информационных систем // Научный сервис в сети Интернет: труды XVIII Всероссийской научной конференции. М.: ИПМ им. М.В. Келдыша, 2016. С. 249-258.
9. Педагогический потенциал: материалы Всеросс. с межд. участ. науч.-практ. конф. / Баранова И.М., Антоненкова О.Е., Камозина О.В., Охлупина О.В. Брянск: БГИТУ, 2023. 337 с.
10. Прокудин Д.Е., Низомутдинов Б.А. Развитие сервисов информационной поддержки научной конференции // Научный сервис в сети Интернет: труды XXIV Всероссийской научной конференции. М.: ИПМ им. М.В. Келдыша, 2022. С. 434-454.
11. Семёнов Е.В., Соколов Д.В. Методологические проблемы комплексных исследований цифровой трансформации научных ком-

- муниаций // Управление наукой: теория и практика. 2021. Том 3, №2. С. 76-98.
12. Скородумова О.Б. Отечественные подходы к интерпретации информационного общества: постиндустриалистская, синергетическая и постмодернистская парадигмы // Информационный гуманитарный портал. Знание. Понимание. Умение. 2009. №4. С. 4.
 13. Gareth Dwyer. Flask by example. Unleash the full potential of the Flask web framework by creating simple yet powerful web applications. 2016. 248 p.
 14. Matthias Biehl. OpenID Connect – End-user Identity for Apps and APIs. API University Press, 2019, 138 p.
 15. Shalabh Aggarwal. Flask Framework Cookbook: Over 80 proven recipes and techniques for Python web development with Flask. 2nd Edition. 2019. 384 p.
 16. Quick start – Flask-RESTX. URL: <https://flask-restx.readthedocs.io/en/latest/quickstart.html> (дата обращения: 11.08.2023)
 17. Quickstart – Requests. URL: <https://requests.readthedocs.io/en/latest/user/quickstart/> (дата обращения: 13.08.2023)
 18. json – JSON encoder and decoder. URL: <https://docs.python.org/3/library/json.html> (дата обращения: 01.08.2023)

References

1. Alan Bewley. *Learning SQL*. (Transl. from English) M.: Symbol-Plus, 2017, 309 p.
2. Barry Pollard. *HTTP/2 in action*. (Transl. from English) M.: DMK-Press, 2021, 424 p.
3. Bogdanova I.F. Nauchnyye kommunikatsii v onlaynovom prostranstve [Scientific communications in the online space]. *Nauka i innovatsii* [Science and innovation], 2014, no. 4(134), pp. 13-16.
4. Golovanova E.V. Prognosticheskiy potentsial teorii postindustrializma D. Bella [The prognostic potential of D. Bell's theory of post-industrialism]. *Vestnik slavyanskikh kul'tur* [Bulletin of Slavic cultures], 2013, no. 4(30), pp. 11-17.

5. Guskov A.E., Vasilkov A.V. Sredstva podderzhki provedeniya nauchnykh konferentsiy: obzor i sravneniye [Tools to support scientific conferences: review and comparison]. *Vestnik Novosibirskogo gosudarstvennogo universiteta. Seriya: Informatsionnyye tekhnologii* [Bulletin of Novosibirsk State University. Series: Information technologies], 2010, vol. 8, no. 4, pp. 35-45.
6. Zadorozhny S.S., Fadeev E.P. *Ob'yektno-orientirovannoye programirovaniye na yazyke Python* [Object-oriented programming in Python]. M.: Faculty of Physics, Moscow State University M.V. Lomonosova, 2022, 40 p.
7. Kolisnichenko D.N. *PHP i MySQL. Razrabotka veb-prilozheniy* [PHP and MySQL. Web application development]. St. Petersburg: Publishing house BHV-Petersburg, 2009, 607 p.
8. Mbogo I.A., Prokudin D.E., Chugunov A.V. Razrabotka instrumentov integratsii nauchnoy informatsii v prostranstve raznorodnykh informatsionnykh sistem [Development of tools for integrating scientific information in the space of heterogeneous information systems]. *Nauchnyy servis v seti Internet: trudy XVIII Vserossiyskoy nauchnoy konferentsii* [Scientific service on the Internet: proceedings of the XVIII All-Russian Scientific Conference], M.: IPM im. M.V.Keldysh, 2016, pp. 249-258.
9. *Pedagogicheskiy potentsial: materialy Vserossiyskoy s mezhdunarodnym uchastiyem nauchno-prakticheskoy konferentsii* [Pedagogical potential: materials of the All-Russian scientific and practical conference with international participation] / Baranova I.M., Antonenkova O.E., Kamoziina O.V., Okhlupina O.V. Bryansk: BGITU, 2023, 337 p.
10. Prokudin D.E., Nizomutdinov B.A. Razvitiye servisov informatsionnoy podderzhki nauchnoy konferentsii [Development of information support services for scientific conferences]. *Nauchnyy servis v seti Internet: trudy XVIII Vserossiyskoy nauchnoy konferentsii* [Scientific service on the Internet: proceedings of the XXIV All-Russian Scientific Conference]. M.: IPM im. M.V.Keldysh, 2022, pp. 434-454.
11. Semenov E.V., Sokolov D.V. Metodologicheskiye problemy kompleksnykh issledovaniy tsifrovoy transformatsii nauchnykh kommu-

- niuatsiy [Methodological problems of comprehensive research into the digital transformation of scientific communications]. *Upravleniye naukoy: teoriya i praktika* [Management of science: theory and practice], 2021, vol. 3, no. 2, pp. 76-98.
12. Skorodumova O.B. Otechestvennyye podkhody k interpretatsii informatsionnogo obshchestva: postindustrialistskaya, sinergeticheskaya i postmodernistskaya paradigm [Domestic approaches to the interpretation of the information society: post-industrialist, synergetic and postmodern paradigms]. *Informatsionnyy gumanitarnyy portal. Znaniye. Ponimaniye. Umeniye* [Electronic magazine. Knowledge. Understanding. Skill], 2009, no. 4, p. 4.
 13. Gareth Dwyer. *Flask by example. Unleash the full potential of the Flask web framework by creating simple yet powerful web applications*. PACKT publishing, 2016. 248 p.
 14. Matthias Biehl. *OpenID Connect – End-user Identity for Apps and APIs*. API University Press, 2019, 138 p.
 15. Shalabh Aggarwal. *Flask Framework Cookbook: Over 80 proven recipes and techniques for Python web development with Flask*. 2nd Edition, 2019, 384 p.
 16. *Quick start – Flask-RESTX*. <https://flask-restx.readthedocs.io/en/latest/quickstart.html> (accessed August 11, 2023)
 17. *Quickstart – Requests*. <https://requests.readthedocs.io/en/latest/user/quickstart/> (accessed August 13, 2023)
 18. *json – JSON encoder and decoder*. <https://docs.python.org/3/library/json.html> (accessed August 01, 2023)

ДАНИЕ ОБ АВТОРАХ

Камозина Олеся Владимировна, доцент кафедры «Математика», кандидат физико-математических наук, доцент
Брянский государственный инженерно-технологический университет
проспект Станке Димитрова, 3, г. Брянск, 241037, Российская Федерация
ovkamosina@yandex.ru

Охлупина Ольга Валентиновна, доцент кафедры «Математика», кандидат физико-математических наук, доцент
Брянский государственный инженерно-технологический университет
проспект Станке Димитрова, 3, г. Брянск, 241037, Российская Федерация
helga131081@yandex.ru

Маганков Кирилл Сергеевич, студент
Брянский государственный инженерно-технологический университет
проспект Станке Димитрова, 3, г. Брянск, 241037, Российская Федерация
kirill.magankov@gmail.com

Рябцев Николай Павлович, студент
Брянский государственный инженерно-технологический университет
проспект Станке Димитрова, 3, г. Брянск, 241037, Российская Федерация
lolofmeistahhz@gmail.com

DATA ABOUT THE AUTHORS

Olesya V. Kamozina, Associate Professor, Department of Mathematics, Candidate of Physical and Mathematical Sciences
Bryansk State Engineering and Technology University
3, Stanke Dimitrova Ave., Bryansk, 241037, Russian Federation
ovkamoziina@yandex.ru
ORCID: <https://orcid.org/0000-0003-2803-6016>

Olga V. Okhlupina, Associate Professor, Department of Mathematics, Candidate of Physical and Mathematical Sciences
Bryansk State Engineering and Technology University

*3, Stanke Dimitrova Ave., Bryansk, 241037, Russian Federation
helga131081@yandex.ru
ORCID: <https://orcid.org/0000-0003-0694-2990>*

Kirill S. Magankov, Student

*Bryansk State Engineering and Technology University
3, Stanke Dimitrova Ave., Bryansk, 241037, Russian Federation
kirill.magankov@gmail.com*

Nikolay P. Ryabtsev, Student

*Bryansk State Engineering and Technology University
3, Stanke Dimitrova Ave., Bryansk, 241037, Russian Federation
lolofmeistahz@gmail.com*

Поступила 13.04.2024

После рецензирования 06.05.2024

Принята 15.05.2024

Received 13.04.2024

Revised 06.05.2024

Accepted 15.05.2024