Министерство науки и высшего образования Российской Федерации

федеральное государственное бюджетное образовательное учреждение   
высшего образования

"Алтайский государственный технический университет

им. И. И. Ползунова"

Факультет информационных технологий

Кафедра Информатики, вычислительной техники

и информационной безопасности

Отчет защищен с оценкой\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.И. Попкова

(подпись руководителя от вуза) (инициалы, фамилия)

“\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2021 г.

ОТЧЕТ

о производственной\_практике

(вид практики)

в ООО «Научно-техническое предприятие Специальная электроника»

(название предприятия, организации, учреждения)

ПрП\_09.03.01.13.000\_ОТ

(обозначение документа)

Студент группы ИВТ-71\_\_\_А.В. Комаров

(инициалы, фамилия)

Руководитель практики

от предприятия директор ООО «Научно-техническое предприятие

Специальная электроника» к.т.н. С.Ю. Тырышкин­­­

(должность, ученое звание) (инициалы, фамилия)

Руководитель практики

от университета доцент каф. ИВТ и ИБ, к.п.н. А.И. Попкова

(должность, ученое звание) (инициалы, фамилия)

2021

Министерство науки и высшего образования Российской Федерации

федеральное государственное бюджетное образовательное учреждение   
высшего образования

"Алтайский государственный технический университет

им. И. И. Ползунова"

Факультет информационных технологий

Кафедра Информатики, вычислительной техники

и информационной безопасности

Отчет защищен с оценкой\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.И. Попкова

(подпись руководителя от вуза) (инициалы, фамилия)

“\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2021 г.

ОТЧЕТ

о производственной\_практике

(вид практики)

в ФГБОУ ВО АлтГТУ

(название предприятия, организации, учреждения)

ПрП\_09.03.01.13.000\_ОТ

(обозначение документа)

Студент группы ИВТ-71\_\_\_А.В. Комаров

(инициалы, фамилия)

Руководитель практики

от предприятия УКАЗЫВАЕМ НАУЧНОГО РУКОВОДИТЕЛЯ ВКР!!!­­­

(должность, ученое звание) (инициалы, фамилия)

Руководитель практики

от университета доцент каф. ИВТ и ИБ, к.п.н. А.И. Попкова

(должность, ученое звание) (инициалы, фамилия)

2021

ФГБОУ ВО «Алтайский государственный технический университет им. И. И. Ползунова»

Кафедра «Информатика, вычислительная техника и информационная безопасность»

Индивидуальное задание

на производственную практику

(технологическая (вторая производственная практика)) студенту 4 курса Комарову А. В. группы ИВТ-71

Профильная организация: ИЗ ПРИКАЗА О ПРАКТИКЕ

Сроки практики: 03.05.2021 г. - 17.05.2021 г.

Тема: «Разработка информационного ресурса для организации и проведения научных конференций»

**Рабочий график (план) проведения практики:**

|  |  |  |  |
| --- | --- | --- | --- |
| **№**  **п/п** | **Содержание**  **раздела**  **(этапа) практики** | **Сроки**  **выполнения** | **Планируемые результаты**  **практики** |
| 1 | Получить индивидуальное задание на практику.  Пройти инструктаж по охране труда и технике безопасности.  Обоснование актуальности тематики ВКР  Написание 1 главы бакалаврской работы | 1 неделя | Базовое формирование компетенций:  **ОК-5:** осуществлять обмен информацией в устной и письменной формах на русском и иностранном языках  **ОК-6:** работать в коллективе, толерантно воспринимая социальные, этнические, конфессиональные и культурные различия.  **ОК-7:** развить навыкисамоорганизации и самообразования.  **ОПК-5:** решение задач сбора, изучения и систематизации информации по тематике будущей ВКР на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности. |
| 2 | Инсталляция и настройка современного программного обеспечения, необходимого для проектирования объекта профессиональной деятельности;  Использование современных технологий проектирования и реализации программно-аппаратного обеспечения по тематике ВКР;  Самостоятельная практическая реализация технологий проектирования, конструирования, исследования объектов профессиональной деятельности по теме ВКР  Написание 2 главы бакалаврской работы | 2 неделя | Базовое формирование компетенций:  **ОПК-1:. с**пособность инсталлировать программное и аппаратное обеспечение для информационных и автоматизированных систем, необходимых для выполнения задания по практике.  **ОПК-2:** освоение методик использования программных средств для решения поставленной практической задачи.  **ОПК-4:** настройка программно-аппаратных комплексов.  **ПК-1:** разработка интерфейса, базы данных, реализация программно-аппаратных средств  **ПК-2:** разработка компонентов аппаратно-программных комплексов и баз данных, используя современные инструментальные средства и технологии программирования.  **ПК-3:** обоснование проектных решений, проверка их корректности и эффективности.  **ПК-5:** сопряжение аппаратных и программных средств, в том числе самостоятельно разработанных.  **ПК-6:** подключение и настройка модулей ЭВМ и периферийного оборудования, необходимого для выполнения задания по практике. |
| 3 | Оформить и сдать отчет по практике. В отчете отразить полученные результаты | 2 неделя | Базовое формирование компетенции  ОК-5: оформление результатов  работы. |
| Руководитель практики от вуза \_\_\_\_\_\_\_\_\_\_\_\_ Попкова А.И., к.п.н., доцент каф. ИВТиИБ  (подпись) (Ф.И.О., должность) | | | |  |
| Руководитель практики от  профильной организации С ТИТУЛЬНОГО ЛИСТА!!!  (подпись) (Ф.И.О., должность) | | | |  |

Задание принял к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Комаров А. В.

(подпись) (Ф.И.О.)

**Инструктаж по ОТ, ТБ, ПБ, ПВТР**

Инструктаж обучающегося по ознакомлению с требованиями охраны труда, техники безопасности, пожарной безопасности, а также правилами внутреннего трудового распорядка проведен «\_03\_» \_мая\_\_ 2021 г.

Руководитель практики от

профильной организации \_\_\_\_\_\_\_\_\_\_\_ С ТИТУЛЬНОГО ЛИСТА!!!!!

(подпись)

МП

**Оглавление**

[Введение 7](#_Toc73326428)

[1 Описание и анализ материалов по проведению научных конференций 8](#_Toc73326429)

[**1.1 Общие сведения об организации конференций** 8](#_Toc73326430)

[**1.2 Проблемы организации научных конференций** 9](#_Toc73326431)

[**1.3 Обзор существующих решений для публикации статей** 11](#_Toc73326432)

[**1.3.1 Международная научно-техническая конференция "Измерение, контроль, информатизация"** 12](#_Toc73326433)

[**1.3.2 МОЛОДЕЖЬ И СОВРЕМЕННЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ** 13](#_Toc73326434)

[**1.3.3 ЛОМОНОСОВ** 16](#_Toc73326435)

[**1.4 Выбор целей и задач при разработке информационного ресурса для проведения научных конференций** 19](#_Toc73326436)

[2 Выбор средств разработки и проектирование ресурса для проведения научных конференций 21](#_Toc73326437)

[**2.1 Определение основных требований к разрабатываемому ресурсу** 21](#_Toc73326438)

[**2.1.1 Функциональные требования** 21](#_Toc73326439)

[**2.1.2 Архитектура** 27](#_Toc73326440)

[**2.1.3 Мощность** 27](#_Toc73326441)

[**2.1.4 Защита информации** 27](#_Toc73326442)

[**2.2 Выбор среды и средств разработки** 29](#_Toc73326443)

[**2.2.1 Среда разработки** 29](#_Toc73326444)

[**2.2.2 Технологии для разработки визуальной части ресурса** 30](#_Toc73326445)

[**2.2.3 Технологии для разработки серверной части** 33](#_Toc73326446)

[**2.2.4 Хранение данных** 33](#_Toc73326447)

[**2.3 Проектирование базы данных** 34](#_Toc73326448)

[**2.3.1 Структура базы данных** 34](#_Toc73326449)

[**2.3.2 Описание сущностей** 34](#_Toc73326450)

[**2.3.3 Описание связей** 40](#_Toc73326451)

[3 Программная реализация ресурса для проведения научных конференций 44](#_Toc73326452)

[**3.1 Структура web-приложения** 44](#_Toc73326453)

[**3.2 Описание методов REST api** 44](#_Toc73326454)

[**3.3 Тестирование разработанного продукта** 44](#_Toc73326455)

[Вывод 45](#_Toc73326456)

[Список использованных источников – отступы!!! 46](#_Toc73326457)

Введение

В современном мире наука занимает важное место в жизни всего человечества. Огромное количество людей начинает все больше и больше интересоваться теми или иными предметами науки, научными журналами и шоу. Именно из-за этого в последнее время происходит популяризация науки среди самых различных слоев населения. Существует большое количество различных конференций, затрагивающих совершенно разные тематики. Такой темп развития возможен лишь благодаря одному мощному инструменту, позволяющему свободно искать и распространять информацию. Этим инструментом является интернет.

Аудитория интернета насчитывает более 4 миллиардов человек по всему миру, и это число продолжает увеличиваться с каждым днем [13]. Веб-серфинг для современного человека стал таким же привычным элементом жизни, как, например, поход в магазин. Даже поиск каких-либо организаций обычно начинается с поиска их веб-сайта. Компании, не имеющие веб-ресурсы, несут огромные убытки, поэтому, в конечном счете, возникает необходимость в создании хотя бы небольшого корпоративного сайта [12].

Различные веб-приложения, облачные сервисы и огромное число порталов становятся все популярнее с каждым днём. Именно поэтому отличным решением будет разработать интернет-ресурс для проведения научных конференций и публикации статей. Создание веб-сайта позволит обеспечить постоянный приток новых пользователей, а также будет являться действительно удобным и приоритетным способом получения информации среди населения.

1 Описание и анализ материалов по проведению научных конференций

## **1.1 Общие сведения об организации конференций**

Для того, чтобы построить модель предметной области необходимо проанализировать потребности и имеющиеся для их удовлетворения ресурсы, что позволит составить точное техническое задание. Благодаря нему получится избежать возможных проблем, с которыми можно столкнуться в процессе разработки [8].

По определению научной конференцией называется организация научной деятельности, в которой исследователи представляют и обсуждают свои работы [1].

Обычно конференции проводит заранее сформированный организационный комитет, который пишет информационное письмо. В письме указываются необходимые участникам конференции данные, такие как название конференции, разделы и тематика, сроки проведения, цена организационных взносов, дата окончания приема публикаций, а также требования к их оформлению.

Далее письмо, содержащее основную информацию о конференции, отсылается потенциальным участникам. Они в свою очередь составляют тезисы и отправляют их обратно в организационный комитет для проверки.

Организационный комитет либо принимает работы и, после оплаты организационного взноса, добавляет в сборник, либо отклоняет, если какие-либо из условий были не соблюдены.

По окончании приема тезисов проводится заседание, в котором все участники, подавшие свои работы, выступают. По завершении выступлений подводятся итоги и выбираются призеры, которые должны будут получить дипломы.

Далее проводится верстка сборника тезисов, который в конечном счете и публикуется, а в случаях, если сборник будет индексироваться, то уже будет происходить его индексация в описанных базах данных, таких как, РИНЦ, SCOPUS и др.

Порядок проведения конференций представлен на рисунке 1.1.

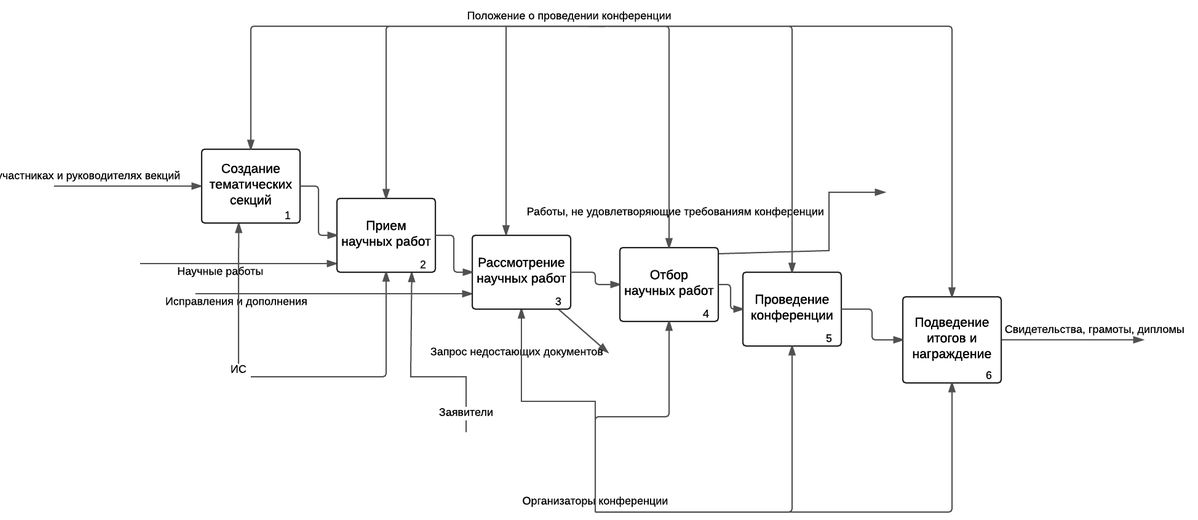


Рисунок 1.1 – Порядок проведения научной конференции

## **1.2 Проблемы организации научных конференций**

Сайту конференции, конечно, необходимо хранить огромные объемы данных, связанных с участниками, их работами, научными руководителями, а также иметь все данные по проведенным конференциям за несколько лет.

Потребность в хранении больших объемов информации и эффективной работе с ними призывает к поиску новых методов обработки данных. Одним из таких методов безопасного хранения, доступа, анализа и изменения информации являются базы данных [11].

С повышением уровня информатизации и появлением доступных ЭВМ стоимость хранения информации на электронных носителях стала меньше той же информации на бумажных носителях, в то время как эффективность первого способа при обработке данных является на порядок большей. Исходя из этого факта, большинство государственных и коммерческих организаций успешно применяют базы данных для работы с информацией, связанной с их предметной областью.

Эффективность работы с базами данных заключается не только в экономической выгоде, но и в гибкости разработки таковых для каждой из предметных областей [3]. База данных может учитывать определенные правила для хранимой информации, обеспечивать выбор необходимой информации в малые сроки, обеспечивать безопасность доступа к информации и её хранения и многое другое.

Однако с хранимыми данными необходимо взаимодействовать и как раз для доступа к базам данных, их контролю и полному администрированию, существуют специальные комплексы программно-языковых средств, называемые – системы управления базами данных или сокращенно СУБД [4]. В мире существует не одна СУБД, каждая из них имеет свои преимущества и недостатки, но практически все находится в процессе постоянного совершенствования и улучшения. Без этих особых средств работа с базами данных была бы невозможна.

Однако при использовании любой СУБД процесс добавления тезисов будет неудобен для организационного комитета, так как все статьи будут добавляется в базу данных вручную, с помощью выполнения написанного запроса. Однако данный процесс было бы гораздо удобнее автоматизировать таким образом, чтобы пользователи сами загружали необходимую информацию, а после завершения срока приема тезисов все статьи со статусом “Принята” заносились на страницу сборника автоматически. Это позволило бы упростить работу организационного комитета и из его вмешательства остались только проверка и изменение статуса заявок участников конференции.

Для реализации такого подхода и оптимизации ресурсов хорошим выбором будет вынесение всех работ с базой данных на сервер, который в свою очередь будет лишь принимать запросы и отвечать на них.

Данный подход называется клиент-серверная архитектура и в нем сайт, открытый у пользователя, будет общаться с сервером, который уже и будет выполнять запросы к базе данных. Это позволит разгрузить базу данных и ускорить процесс получения информации (Рисунок 1.2).



Рисунок 1.2 – Клиент-серверная архитектура

Ещё в приложении должна быть возможность легко оповещать предыдущих участников о предстоящих конференциях, о статусах поданных ими тезисов и статусах оплаты организационных взносов, если таковые будут иметься. Названия разделов могут изменяться, о чем также будет необходимо оповещать участников. Все эти моменты необходимо учесть и реализовать данный функционал, чтобы пользователь постоянно получал актуальную информацию, необходимую ему.

## **1.3 Обзор существующих решений для публикации статей**

На данный момент существует большое количество ресурсов, позволяющих публиковаться во всевозможных журналах и участвовать в конференциях. Однако в большинстве своем данные ресурсы проводят лишь те конференции, которые они сами и организуют. Это создает определенное неудобство для участников, поскольку для отслеживания актуальной информации им приходится посещать большое число ресурсов.

Также существуют и такие сайты, которые позволяют организовывать свои собственные конференции. Подобные ресурсы представляют собой обычную площадку, благодаря которой участник оставляет заявки и отправляет работы на участие организационному комитету, которые и проводит конференцию. Участникам такая реализация позволяет с одного ресурса подавать заявки на многочисленные конференции, представленные на них.

Существующие варианты решают большое количество организационных проблем, однако у них имеются и свои недостатки. Одними из них являются сложность проверки оплаты отправленных публикаций, неудобный процесс отправления заявок на проведение собственных конференций. Большая часть существующих решений имеет непонятный и сложный дизайн, который может отпугивать потенциальных участников, а также затрудняет работу организационного комитета.

### **1.3.1 Международная научно-техническая конференция "Измерение, контроль, информатизация"**

На данный момент Алтайский государственный технический университет им. И.И. Ползунова имеет свой ресурс [6]. Его главная страница представлена на рисунке 1.3. Дизайн уже не является актуальным, а основной акцент не направлен на интересующие заявителей блоки.



Рисунок 1.3 – Главная страница сайта mca.altstu.ru

Также недочетом можно выделить тот факт, что на данном сайте проводится лишь одна конференция, хотя небольшое расширение функционала позволило бы оптимизировать ресурсы университета.

Форма отправления тезисов представлена на рисунке 1.4. Она перегружена элементами интерфейса, некоторые из которых используются пользователями крайне редко. Это не позволяет сфокусироваться на основных элементах ввода, и заявитель тратит свое время, чтобы разобраться в процессе отправки. Это делает процесс добавления тезисов сложным и неудобным.

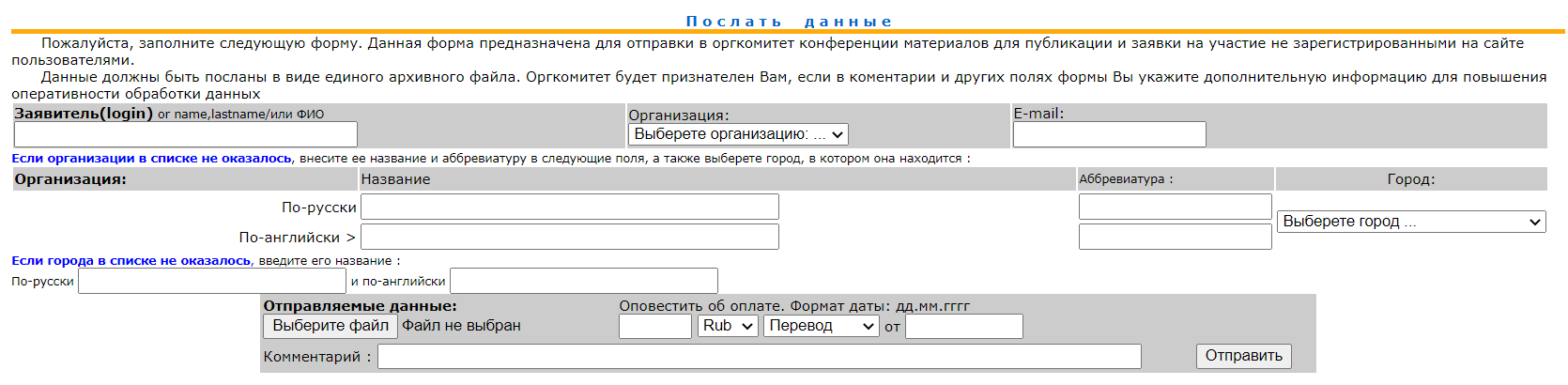


Рисунок 1.4 – Форма отправки публикаций

### **1.3.2 МОЛОДЕЖЬ И СОВРЕМЕННЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ**

Хорошим примером ресурса будет сайт Томского политехнического университета, проводящего конференцию “МОЛОДЕЖЬ И СОВРЕМЕННЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ” [5]. Его главная страница (Рисунок 1.5) имеет большое число различных кнопок и ссылок, нагружающих интерфейс. Большая часть информации представлена в PDF, которые скачиваются при клике на кнопки. Это вызывает определенное неудобство для пользователя, так как для того, чтобы ознакомиться с важной информацией, например, с секциями конференции ему приходится скачивать документ и открывать его. В последующем, после посещения данного сайта придется удалять огромное число загруженных файлов.

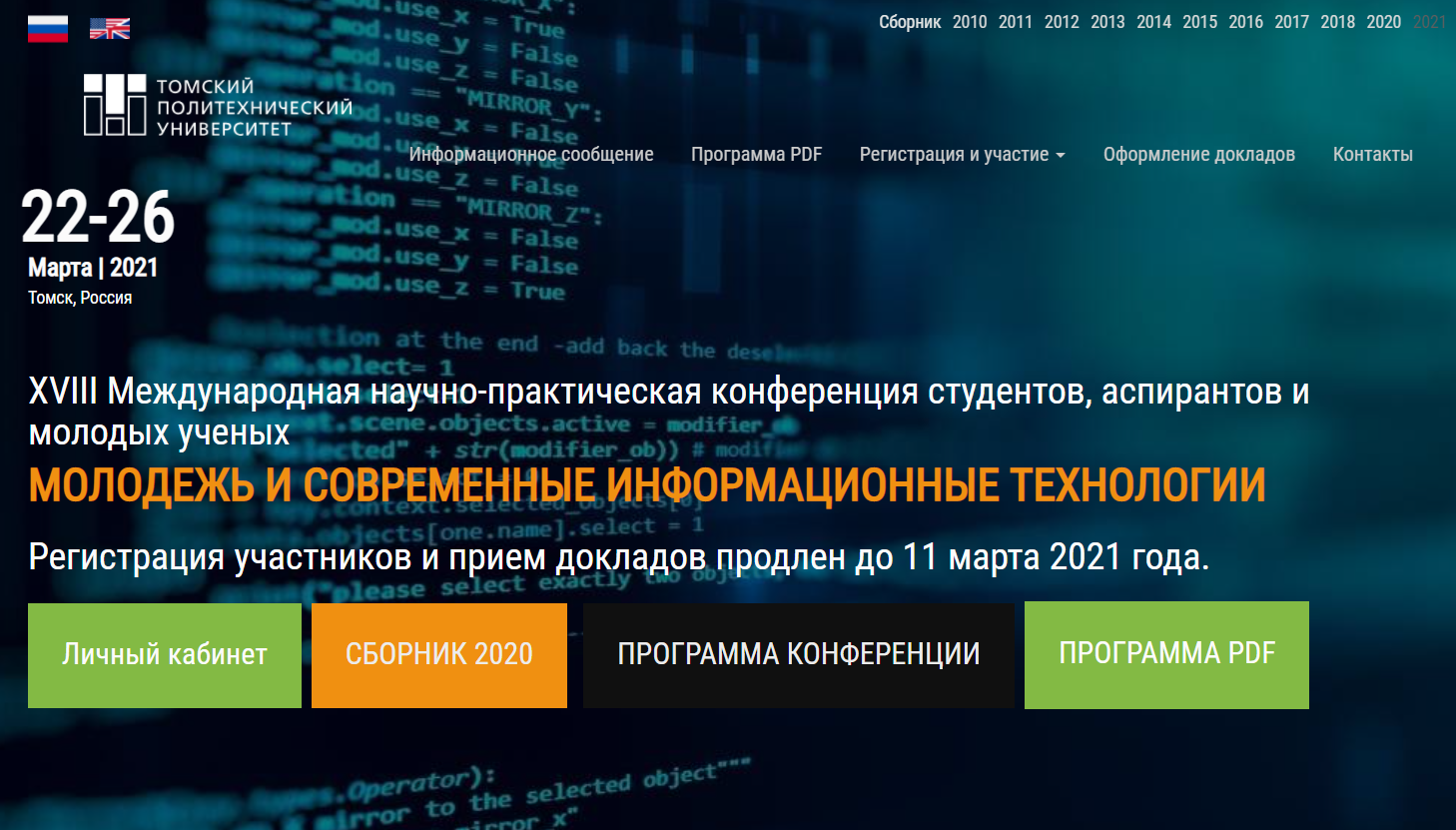


Рисунок 1.5 – Главная страница сайта конференции “МОЛОДЕЖЬ И СОВРЕМЕННЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ”

Также неудобством для пользователя будет являться мобильная версия данного сайта (Рисунок 1.6). Как можно увидеть, большая часть блоков сайта смещается и данный сайт в принципе можно назвать адаптивным, однако некоторые блоки перегораживают другие, что не позволяет кликать по ссылкам, или читать текст.

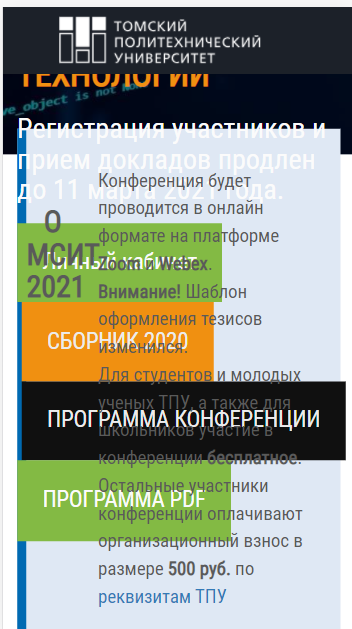


Рисунок 1.6 – Мобильная версия сайта конференции “МОЛОДЕЖЬ И СОВРЕМЕННЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ”

Однако несмотря на недостатки данного сайта, на нем удобно реализована возможность просмотра конференции в zoom или webex, с помощью кнопок, направляющих на данные ресурсы (Рисунок 1.7). Для конференций, приводящихся удаленно это не сложное, но достаточно удобное решение.

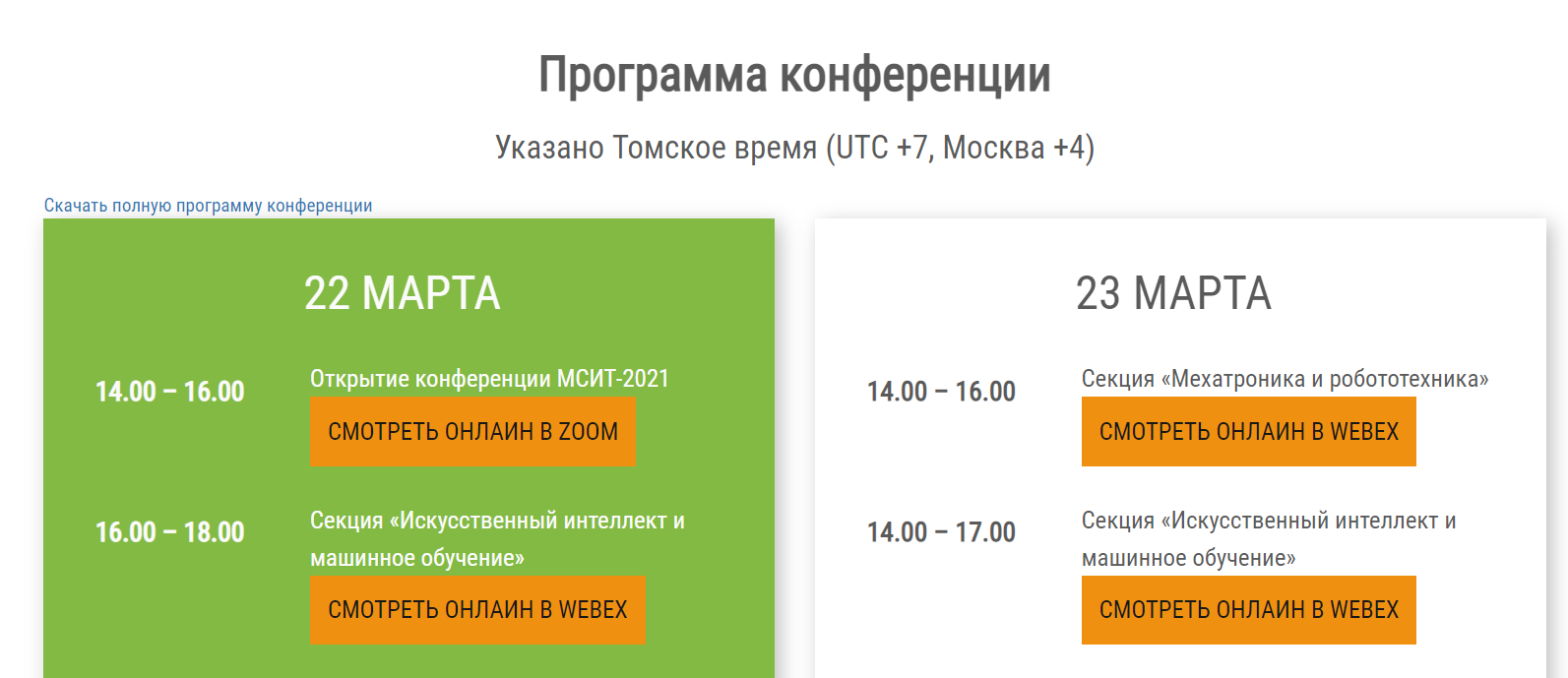


Рисунок 1.7 – Возможность просмотра конференции удаленно

### **1.3.3 ЛОМОНОСОВ**

Единственным ресурсом, позволяющим регистрировать свои собственные конференции, является “ЛОМОНОСОВ” [9]. Как можно увидеть по его главной странице (Рисунок 1.8) дизайн достаточно простой и удобный. Регистрироваться на конференции просто, однако несмотря на это им есть что доработать, например, в формах иногда требуется как лишняя информация, так иногда полей и недостает.

Все дело в том, что все формы формирует сам организатор конференции, хотя некоторый базовый список данных должен иметься изначально, это бы упростило работу с данным ресурсом и уменьшило бы время создания своих мероприятий.

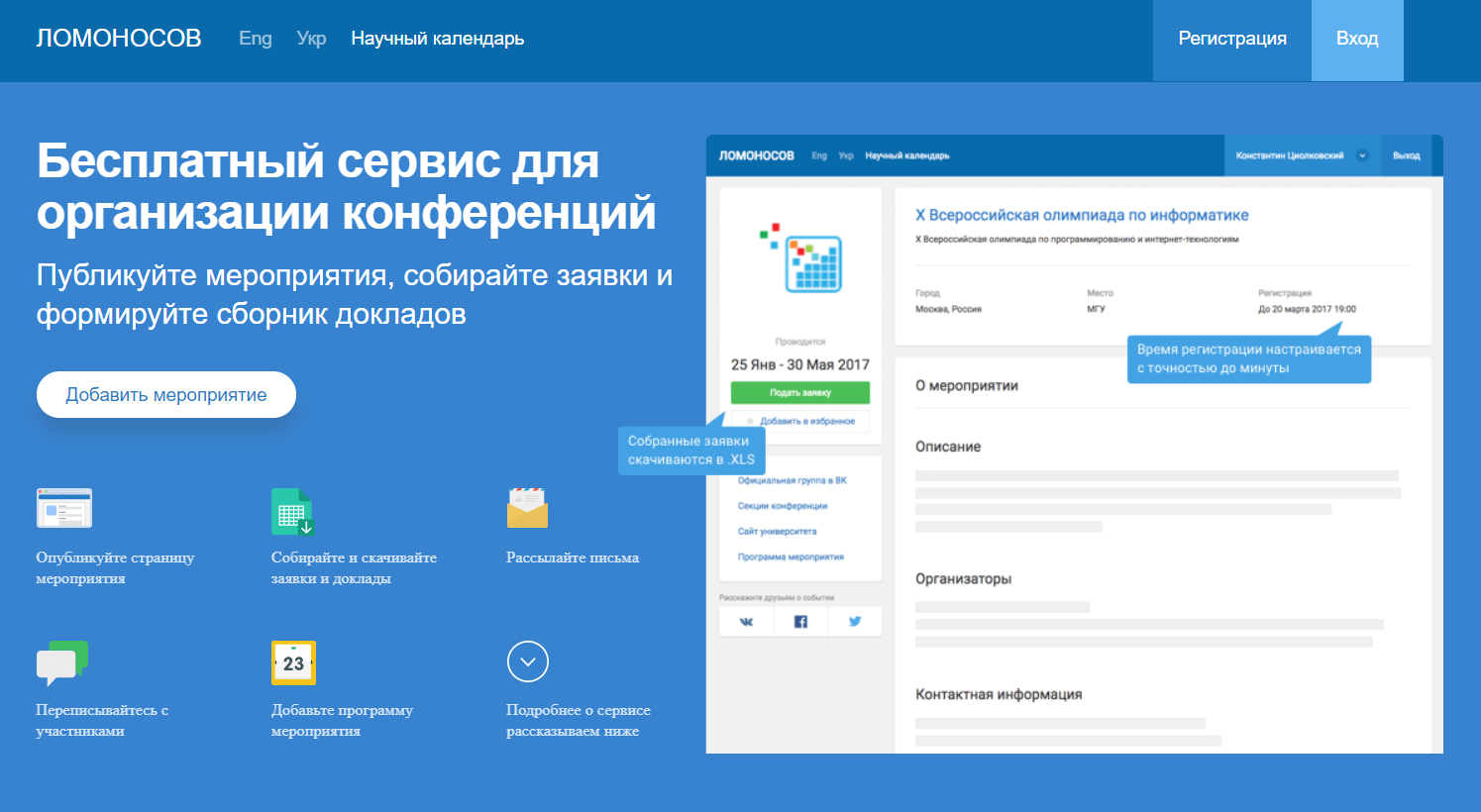


Рисунок 1.8 – Главная страница интернет -ресурса “ЛОМОНОСОВ”

Плюсом данного ресурса также можно назвать адаптивный дизайн (Рисунок 1.9), который легко реализуется с помощью огромного числа удобных инструментов разработки. Это не займет много времени, зато улучшит пользовательский опыт и позволит проверять актуальную информацию в любом месте и с любого гаджета.

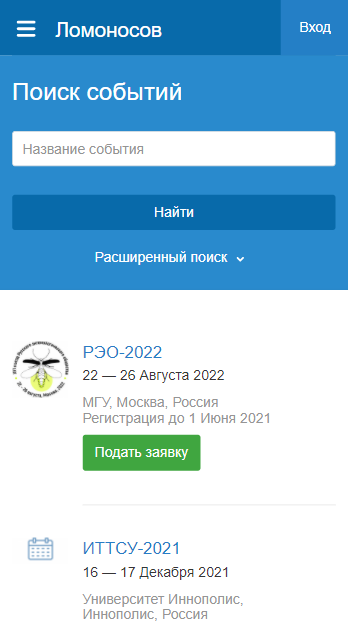


Рисунок 1.9 – Мобильная версия интернет-ресурса “ЛОМОНОСОВ”

Главным недочетом данного сервиса является отсутствие каких-либо данных, после проведения конференции. Например, он не предоставляет доступ сборников уже проведенных конференций, а также отсутствует информация о призерах. Есть проблема с получением уведомлений, содержащих актуальную информацию о проведения мероприятия.

## **1.4 Выбор целей и задач при разработке информационного ресурса для проведения научных конференций**

Целью данной работы является разработка удобного и простого информационного ресурса для организации и проведения научных конференций, который позволял бы организационному комитету, модераторам, а также конечным пользователям без проблем пользоваться разработанным ресурсом. Соответственно, данный сайт должен обеспечивать удобство пользования, легкие для заполнения, не перегруженные формы. Данный ресурс должен иметь обширный функционал и простой дизайн, предоставляющий удобное взаимодействие между всеми пользователями данной площадки. Доступ к основной информации должен быть возможен с любой страницы. Многие процессы, например, выгрузка списка принятых работ в базу данных должны быть автоматизированы, чтобы сократить время работы и упростить пользование сайтом, как для организационного комитета, так и для конечного потребителя.

Для достижения цели требуется решить следующие задачи:

* Изучить информацию о предметной области ВКР;
* Проанализировать полученную информацию и выделив проблемные части продумать функционал, позволяющий решить их;
* Определить структуру и пользовательский интерфейс;
* Разработать макет сайта;
* Спроектировать концептуальную модель базы данных;
* Определить стек используемых для разработки технологий;
* Создать базу данных;
* Разработать визуальную часть;
* Разработать серверную часть;
* Протестировать разработанный ресурс и исправить возможные недочеты.

Разработанный ресурс позволит упростить подачу заявок, на публикацию научных работ, чем может привлечь большее число людей в данную отрасль. Также данная система может использоваться университетом, как площадка, на которой он будет проводить собственные конференции, не зависев при этом от сторонних ресурсов.

2 Выбор средств разработки и проектирование ресурса для проведения научных конференций

## **2.1 Определение основных требований к разрабатываемому ресурсу**

### **2.1.1 Функциональные требования**

#### **2.1.1.1 Функции сайта**

Описанный сайт должен выполнять следующие функции:

* Регистрация/ авторизация/ восстановление доступа пользователя;
* Отслеживание состояния отправленных работ;
* Удобная подача заявок на публикацию;
* Хранение дипломов пользователя, в случае награждения его работ;
* Скачивание работы, независимо от времени их публикации;
* Сортировка работ по авторам, названиям, годам публикации;
* Возможность обратной связи между пользователями и организаторами.

#### **2.1.1.2 Структура web-сайта**

Для удобного пользования сайтом необходимо предусмотреть следующие страницы на нем:

* Главная;
* Научный календарь;
* Страница конференции;
* Страница подачи заявки;
* Вышедшие сборники;
* Регистрация;
* Личный кабинет;
* Обратная связь.

##### **2.1.1.2.1 Страница “Главная”**

Данная страница будет находится в меню, и любой пользователь всегда сможет перейти на нее в любой момент времени.

Страница будет включать в себя основную информацию о сайте, такую как основная задача этого ресурса, в каких базах будут индексироваться вышедшие публикации научных статей, а также короткую информацию о конференциях, которые будут проведены в ближайшее время, с возможностью перейти на страницу конференции.

С данной страницы пользователь сможет перейти на любую из страниц, указанных в меню, также если пользователь будет авторизован, то он сможет попасть в личный кабинет.

Главная страница будет доступна всем пользователям, вне зависимости от того, зарегистрированы ли они.

##### **2.1.1.2.2 Страница “Научный календарь”**

Страница будет находится в меню, и любой пользователь всегда сможет перейти на нее в любой момент времени.

Данная страница будет включать в себя информацию о всех конференциях, которые планируется провести, а именно дата проведения, дата регистрации и название форума. Также здесь будет поисковая строка, для возможности поиска определенной конференции, по датам проведения, по названию, по датам регистрации, по этим же критериям будет доступна и сортировка.

С данной страницы пользователь сможет перейти на любую из страниц, указанных в меню, также если пользователь будет авторизован, то он сможет попасть в личный кабинет.

Научный календарь будет доступен всем пользователям, вне зависимости от того, зарегистрированы ли они.

##### **2.1.1.2.3 Страница конференции**

Данная страница будет доступна с двух страниц, со страницы “Научный календарь” и с главной, также на страницы конференций, на которые авторизированный пользователь уже подал заявки, он сможет попасть из личного кабинета, а именно из пункта “Мои заявки”.

Страница будет иметь одинаковый дизайн, но иметь разное наполнение, в зависимости от выбранной конференции. На этой странице будет располагаться вся основная информация о конференции, а именно требования к докладам или тезисам докладов, требования к оформлению, разделы сборника, дата и формат проведения, программа конференции, цена и возможные способы оплаты, информация об организаторах, а также контакты для связи с ними. Помимо информации, данная страница должна содержать кнопку “Подать заявку”, при клике на которую не авторизированному пользователю будет предложено войти, или зарегистрироваться, а авторизированный попадет на страницу подачи заявки.

С данной страницы пользователь сможет перейти на любую из страниц, указанных в меню, также если пользователь будет авторизован, то он сможет попасть в личный кабинет, либо на страницу подачи заявки.

Страница конференции будет доступна всем пользователям, вне зависимости от того, зарегистрированы ли они.

##### **2.1.1.2.4 Страница подачи заявки**

Данная страница будет доступна лишь со страницы конференции.

Она будет представлять из себя форму для отправки заявок, содержащую следующие элементы интерфейса:

* Текстовое поле для ввода ФИО автора; -он же зарегистрирован уже!
* Кнопка для возможности добавления соавтора, которое будет добавлять текстовые поля при клике, на нее;
* Кнопка для возможности удаления соавтора, которое будет удалять последнее текстовое поле при клике, на нее;
* Текстовое поле для ввода ФИО соавтора (их число управляется кнопками);
* Текстовое поле для ввода телефона для связи;
* Текстовое поле для ввода e-mail для связи;
* Раскрывающийся список с возможностью ввода/поиска для выбора университета;
* Текстовое поле для ввода названия статьи;
* Раскрывающийся список с возможностью поиска для выбора раздела;
* Кнопка для загрузки файла доклада;
* Кнопка для загрузки файла подтверждения оригинальности доклада;
* Кнопка для отправления заявки.

После отправления заявки перед пользователем появляется кнопка для оплаты, при клике на которую он будет попадать на одну из платежных систем, которая отправит нам запрос о статусе оплаты. В случае, если оплата пройдет успешно заявка пользователя уходит на рассмотрение орг. комитета, а пользователь попадает в личный кабинет, в раздел “Мои заявки”, где он сможет рассматривать статус своей заявки. Также сообщение статуса будет дублироваться на почту для связи, которую он указал в форме, на странице подачи заявки.

Страница подачи заявки будет доступна лишь авторизированным пользователям.

##### **2.1.1.2.5 Страница “Вышедшие сборники”**

Данная страница будет находится в меню, и любой пользователь всегда сможет перейти на нее в любой момент времени.

Она будет включать в себя список всех опубликованных сборников, с их названием и годом выхода. При клике, на один из сборников, в выпадающем окне появляется вся информация о нем, а именно включенные разделы, названия статьи, ее аннотация, а также список авторов. Помимо информации имеется кнопка для скачивания сборника.

Также на странице будет поисковая строка, для возможности поиска определенного сборника, по датам публикации и по названию, по этим же критериям будет доступна и сортировка.

С данной страницы пользователь сможет перейти на любую из страниц, указанных в меню, также если пользователь будет авторизован, то он сможет попасть в личный кабинет.

Страница будет доступна всем пользователям, вне зависимости от того, зарегистрированы ли они.

##### **2.1.1.2.6 Страница “Регистрация”**

Переход на данную страницу будет осуществляться с помощью кнопки “Зарегистрироваться” в правом верхнем углу экрана.

Она будет представлять из себя форму для регистрации, содержащую следующие элементы интерфейса:

* Текстовое поле для ввода ФИО;
* Раскрывающийся список для выбора пола;
* Текстовое поле для ввода дня рождения;
* Раскрывающийся список с возможностью ввода/поиска для выбора города;
* Текстовое поле для ввода телефона для связи;
* Текстовое поле для ввода e-mail для связи; **- есть при вводе статьи**
* Раскрывающийся список с возможностью ввода/поиска для выбора университета;
* Текстовое поле для ввода пароля;
* Текстовое поле для ввода подтверждения пароля;
* Кнопка для завершения регистрации.

После отправления нажатия на кнопку “Зарегистрироваться”, пользователь появляется на странице, на которой он нажал на кнопку регистрации, уже со статусом авторизирован, а в правом верхнем углу вместо кнопок “Войти” и “Зарегистрироваться” будет выпадающий список, с разделами “Настройки”, “Мои заявки” и “Мои дипломы”.

##### **2.1.1.2.7 Страница “Личный кабинет”**

Переход на данную страницу будет осуществляться с помощью клика по одному из разделов в выпадающем списке, в правом верхнем углу экрана. И в зависимости от раздела, по которому было совершено нажатие, будет выбираться один из пунктов меню, в личном кабинете, а именно “Настройки”, “Мои заявки” и “Мои дипломы”.

В разделе “Настройки” будет находиться форма, как при регистрации, которая позволит менять данные, указанные пользователем при регистрации, например, пароль, ФИО, университет, город и т.д.

В разделе “Мои заявки” будут содержаться список ваших заявок, с названием ваших статей, названия конференции, на которую отправлена данная статья, а также статус вашей заявки.

В разделе “Мои дипломы” будут храниться ваши дипломы, если вы будете номинированы за какие-то статьи. Будет написано, за какую именно публикацию вы получили диплом, в какой номинации вы победили, а также в какой именно конференции, также рядом будет кнопка “Скачать”, для его загрузки.

##### **2.1.1.2.8 Страница “Обратная связь”**

Переход на данную страницу будет осуществляться с помощью клика по кнопке “Обратная связь” внизу страницы.

Она будет содержать в себе текстовое поле, для описания каких-либо ошибок, кнопку загрузки файлов, для загрузки скриншотов, а также кнопку для отправки ошибки.

### **2.1.2 Архитектура**

Как уже было сказано, данному web-ресурсу требуется клиент-серверная архитектура, которая сможет обеспечить высокую скорость разработки. Благодаря данной архитектуре серверный код не будет пересекаться с визуальной частью ресурса, а соответственно код будет более читабелен. Благодаря такому четкому разделению код без проблем можно будет отладить и найти возможные ошибки на этапе тестирования.

### **2.1.3 Мощность**

Этот сайт не требует больших вычислительных мощностей от сервера, так как единоразовое посещение не предполагает более 3 тысяч пользователей одновременно. Соответственно поддержка сайта будет обходиться не дорого, что является весомым плюсом.

### **2.1.4 Защита информации**

Web-сайт доступен из Интернета, чем может привлечь внимание злоумышленников. Для получения доступа к конфиденциальным данным, в них содержащихся, применяются разнообразные векторы атак. Ниже мы рассмотрим их и действия, которые необходимо предпринять, чтобы не допустить взлома нашего web-ресурса.

#### **2.1.4.1 SQL-инъекции**

Приложения используют SQL-запросы для того, чтобы получать, добавлять, изменять или удалять данные, например при редактировании пользователем своих личных данных или заполнении анкеты на сайте. При недостаточной проверке данных от пользователя, злоумышленник может внедрить в форму web-интерфейса приложения специальный код, содержащий кусок SQL-запроса. Например, изменить вместе с именем и фамилией баланс своего счета, посмотреть баланс чужого счета, или же, похитить конфиденциальные личные данные.

Эта уязвимость является следствием недостаточной проверки данных, поступающих от пользователя. Это позволяет злоумышленнику «подсунуть», например, в веб-формы, специально подготовленные запросы, которые «обманут» приложение и позволят прочитать или записать нелегитимные данные.

В случае нашего приложения злоумышленники будут иметь выгоду лишь в получении базы паролей всех пользователей, которые чаще всего имеют одинаковый пароль на всех сайтах, где они зарегистрированы. Больше никаких интересующих их данных в ~~нашей~~ базе данных представлено.

Для того, чтобы избежать утечки данных, необходимо внимательно проверять содержимое текстовых полей, экранировать символы, а также приводить поля к нужным типам данных.

#### **2.1.4.2 XSS (Межсайтовый скриптинг)**

Межсайтовый скриптинг – еще одна ошибка валидации пользовательских данных, которая позволяет передать JavaScript код на исполнение в браузер пользователя. Атаки такого рода часто также называют HTML-инъекциями, ведь механизм их внедрения очень схож с SQL-инъекциями, но, в отличие от них, внедряемый код исполняется в браузере пользователя.

Через JavaScript можно изменять данные, расположенные на странице. Например, злоумышленник сможет заменить кнопку оплаты на свою, и пользователь при клике на эту кнопку будет попадать на поддельную страницу оплаты, где после ввода данных своей карты они будут отправляться мошенникам.

Как уже написано выше, эта уязвимость возникает по причине плохой валидации пользовательских данных, а, следовательно, для избегания утечки пользовательских данных следует внимательно проверять введенные пользователем данные в текстовые поля.

## **2.2 Выбор среды и средств разработки**

### **2.2.1 Среда разработки**

В качестве среды разработки было решено использовать Visual Studio Code — бесплатный и имеющий множество дополнений текстовый редактор, который в первую очередь предназначен для создания и отладки современных веб-приложений [15].

Разработан он компанией Microsoft и доступен для операционных систем Windows, MacOS и Linux.

Распространяется данная программа бесплатно, исходный код её доступен на GitHub.

VS Code поддерживает большое количество языков программирования, включает в себя отладчик, средства для работы с Git, подсветку синтаксиса, технологию автодополнения IntelliSense, инструменты для рефакторинга, терминал и многое другое.

VS Code является достаточно гибким инструментом, расширяемым с помощью плагинов, и это является его основным достоинством. Благодаря этому можно выбрать любой язык программирования и для его использования будет достаточно лишь установить некоторые плагины.

Плагины позволяют менять изначальный вид программного кода и интерфейса (Рисунок 2.1), а также отлично справляются с автоформатированием кода, автодополнением и множеством облегчающих программирование функций.

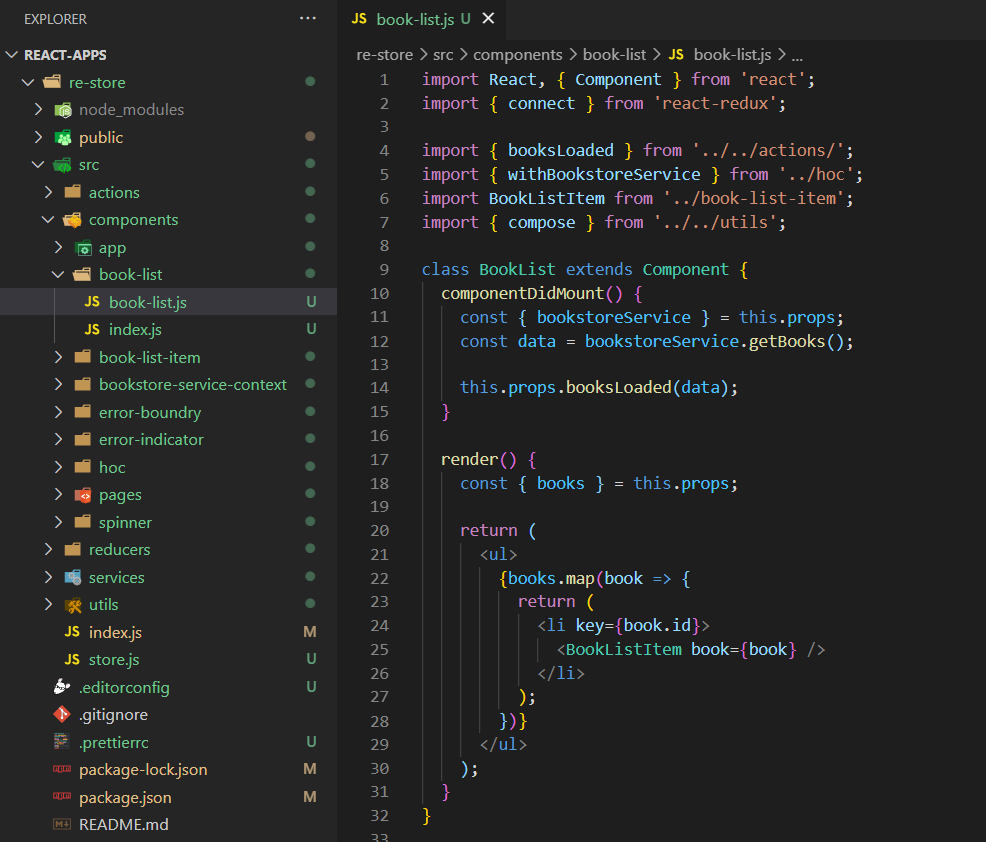


Рисунок 2.1 – Интерфейс Visual Studio Code с использованием плагинов

### **2.2.2 Технологии для разработки визуальной части ресурса**

Основным языком программирования было решено сделать JavaScript с использованием фреймворка React – библиотеки с открытым исходным кодом для разработки пользовательских интерфейсов [7]. Данная библиотека разрабатывается и поддерживается компанией Facebook и огромным сообществом отдельных разработчиков, также свой вклад в данную технологию вносят и другие корпораций, использующие ее.

В основу разработки библиотеки положен компонентный подход, что позволяет создавать компоненты, которые имеют собственное состояние, а после объединять эти компоненты в сложные пользовательские интерфейсы. Поскольку логика самих компонентов реализована на языке программирования JavaScript, можно передавать данные по всему приложению, не привязываясь к интерфейсу DOM (Document Object Model).

Архитектура React построена таким образом, что отрисовка страницы реализована на стороне пользователя, а все отображаемые данные динамически загружаются благодаря асинхронным запросам к серверу. За счет такой архитектуры приложения, написанные на React, работают быстрее, чем обычные многостраничные сайты.

Перемещение по сайту не приводит к перезагрузке страницы. То есть при переходе между страницами, обновляется только часть контента, и не происходит повторная загрузка уже отображенных элементов, таких как меню, логотип и т.д. Это удобно для обычных пользователей, из-за увеличения скорости работы приложений, с которыми они сталкиваются.

Благодаря React используемое клиентом устройство совершает запрос ко всему контенту лишь при первой загрузке. После при перемещении пользователя по сайту большая часть контента уже будет загружена.

Одной из ключевых особенностей библиотеки является возможность использования препроцессора JSX с синтаксисом, схожим с языком разметки HTML, который преобразуется в код языка программирования JavaScript. Препроцессор JSX позволяет описывать структуру интерфейса. Как правило, при использовании библиотеки React для построения компонентов используют именно этот препроцессор, но можно использовать также и чистый JavaScript код.

К особенностям библиотеки также относятся возможность использования однонаправленной передачи данных. Свойства могут передаваться от родительских компонентов к дочерним. Таким образом, компоненты получают свойства как множества постоянных значений. Благодаря этому исключается возможность изменять свойства компонента при передаче данных, но остается возможность изменения свойств через функции обратного вызова. Такой механизм называют «свойства вниз, события наверх» [10].

Библиотека React работает с использованием виртуального интерфейса DOM (Document Object Model). В памяти создается кэш-структура, которая позволяет вычислять разность предшествующего и текущего состояний интерфейса. Таким образом, у разработчика появляется возможность работать со страницей в предположении, что она полностью обновляется, но какие компоненты обновлять, решает сама библиотека.

Также React может быть успешно использован для разработки приложений, которые отлично работают на мобильных устройствах. Такую возможность даёт фреймворк, созданный на базе React – React Native. Он позволяет создавать кроссплатформенные мобильные приложения под наиболее популярные операционные системы iOS и Android. В отличие от многих других кроссплатформенных решений, он использует JavaScript API поверх нативных компонентов. То есть, написав код на JavaScript, можно использовать те же самые фундаментальные стандартные блоки интерфейса пользователя, что и обычные приложения iOS и Android, не имитируя при этом среду браузера. Из-за такого подхода интерфейс будет вести себя точно так же, как и в классическом нативном приложении, написанном на Java или Objective C.

Многие критикуют React из-за того, что сайты, написанные на нем, не индексируются поисковыми системами, потому что приложение загружает информацию по запросу со стороны клиента, а поисковые роботы с трудом могут имитировать данное поведение. Из-за этого некоторые страницы недоступны для сканирования. Однако, используя серверный рендеринг и изоморфное приложение, можно полностью избавится от этой проблемы. Поисковые системы получают готовые HTML страницы со всей метаинформацией и семантической разметкой.

### **2.2.3 Технологии для разработки серверной части**

В качестве серверного языка было принято решение взять PHP, ввиду его популярности и простоты. В настоящее время он является самым распространенным языком программирования, который ориентирован именно на создание web-приложений. Его популярность обусловлена целым рядом преимуществ, которые он дает разработчику:

* простой и понятный синтаксис, позволяющий писать быстро достаточно понятный код;
* слабая динамическая типизация, за счет чего язык позволяет легко оперировать данными разных типов, не боясь ошибки;
* высокая скорость работы и, соответственно, высокая общая производительность ресурсов;
* поддержка огромного числа баз данных.

Все эти факторы обеспечивают быструю скорость программирования и дают хороший потенциал для масштабирования проекта в будущем.

В целом выбор серверного языка не так важен, так как задачи, которые он должен выполнять (Рисунок 2.2), легко решаются любым инструментом, поддерживающим общение с базой данных.

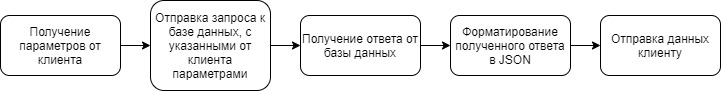


Рисунок 2.2 – Процесс работы сервера

### **2.2.4 Хранение данных**

В нашем случае отличным вариантом будет использовать MySQL [14] по причине того, что она не требует больших вычислительных мощностей, поддерживается уже долгое время, и, соответственно, имеется большое количество специалистов, которые смогут устранить возможные проблемы при их возникновении.

## **2.3 Проектирование базы данных**

### **2.3.1 Структура базы данных**

Результат построения базы данных предметной области представлен на рисунке 2.3.

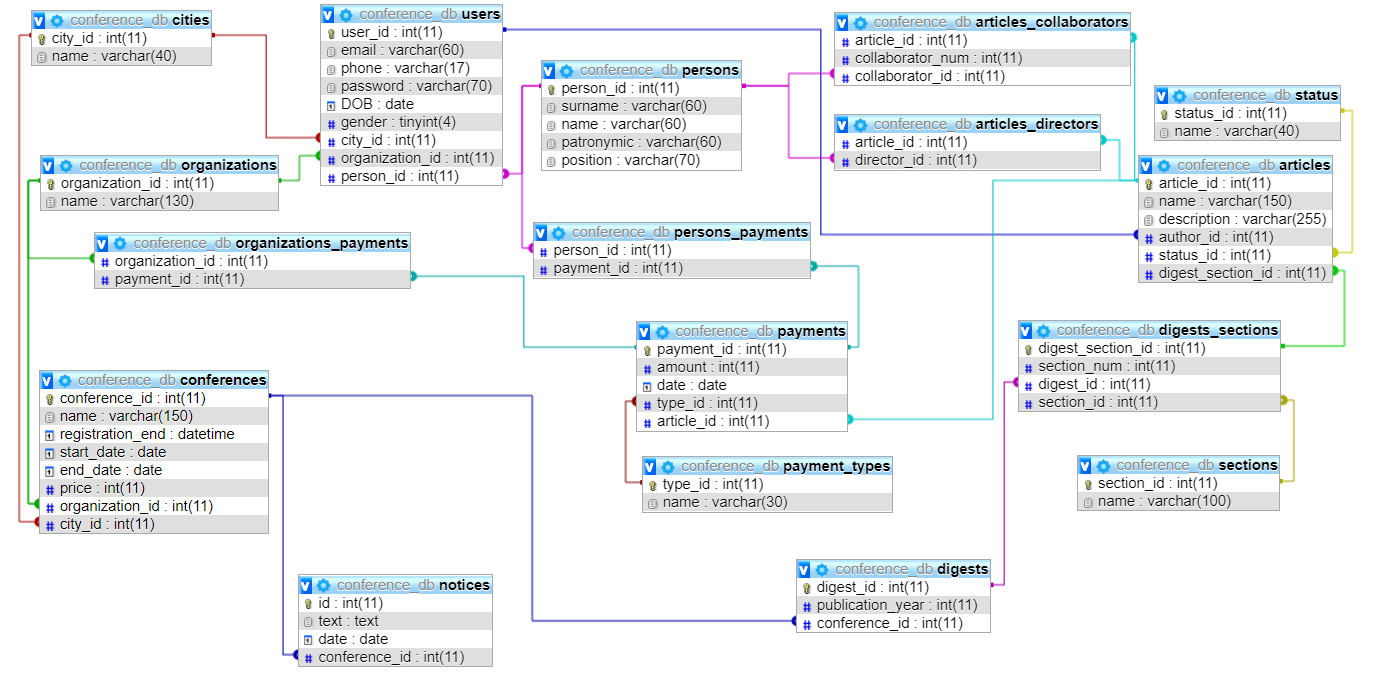


Рисунок 2.3 – База данных предметной области «Сайт конференции»

Нет инфо о степени, должности, звании авторов, в conference - memo

### **2.3.2 Описание сущностей**

1. Сведения о персонах **– А где ID таблицы?**

Включает информацию о данных отдельной персоны, такие как, ФИО и должность

1. Сведения о городе

Включает информацию о названии города

1. Сведения об организации

Включает информацию о названии организации

1. Сведения о пользователях

Включает информацию о данных персоны, дате рождения, пол пользователя, город, организацию, а также email, телефон и пароль.

1. Сведения о статьях

Включают в себя название работы, данные об авторе, описание, статус заявки, а также данные о разделе сборника.

1. Сведения о статусе

Данная сущность создана для хранения различных типов статусов публикации, таких как: “В рассмотрении”, “Принята”, “Не принята” и тд.

1. Сведения о научных руководителях статьи – отметка в поле и все

Данная таблица создана как развязочная, для связи многие ко многим, между сущностями персона и статьи. Она содержит в себе идентификатор статьи и научного руководителя.

1. Сведения о соавторах статьи

Данная таблица создана как развязочная, для связи многие ко многим, между сущностями персона и статьи. Она содержит в себе идентификатор статьи и соавтора, а также номер позиции в списке авторов.

1. Сведения о платежах

Включают в себя сведения о размере платежа, дате его получения и типе платежа.

1. Сведения о типах платежей

Включают в себя сведения о том, как именно отправлены средства.

1. Сведения о платежах персон

Данная таблица создана как развязочная, для связи многие ко многим, между сущностями персона и платежи. Она содержит в себе идентификатор платежа и персоны.

1. Сведения о платежах организаций

Данная таблица создана как развязочная, для связи многие ко многим, между сущностями организация и платежи. Она содержит в себе идентификатор платежа и организации.

1. Сведения о сборнике

Включают в себя сведения о конференции и год публикации сборника.

1. Сведения о разделах

Включают в себя названия разделов.

1. Сведения о разделах сборника

Включают в себя сведения о сборнике, названия разделов выпускаемого сборника, а также их нумерацию.

1. Сведения о конференциях

Включают в себя названия проводимых конференции, дату окончания регистрации, дату проведения, размер оргвзноса, а также город и организацию для проведения.

1. Сведения об объявлениях

Включают в себя текст объявления, а также дату публикации.

Структура рассмотренных сущностей, а именно: название полей, их назначение, а также описание типов данных, приведена в таблицах 1-17.

Нет программы конференции: заезд, порядок выступлений, место проведения, ланчи

Таблица 1 - Структура таблицы “persons”

|  |  |  |
| --- | --- | --- |
| Имя столбца | Назначение | Тип |
| person\_id | Идентификатор персоны | auto increment int |
| surname | Фамилия | varchar(60) |
| name | Имя | varchar(60) |
| patronymic | Отчество | varchar(60) |
| position | Должность | varchar(70) |

Таблица 2 - Структура таблицы “cities”

|  |  |  |
| --- | --- | --- |
| Имя столбца | Назначение | Тип |
| city\_id | Идентификатор города | auto increment int |
| name | Название города | varchar(40) |

Таблица 3 - Структура таблицы “organizations”

|  |  |  |
| --- | --- | --- |
| Имя столбца | Назначение | Тип |
| organization\_id | Идентификатор организации | auto increment int |
| name | Название организации | varchar(130) |

Таблица 4 - Структура таблицы “users”

|  |  |  |
| --- | --- | --- |
| Имя столбца | Назначение | Тип |
| user\_id | Идентификатор пользователя | auto increment int |
| email | Адрес почты пользователя | varchar(60) |
| phone | Номер телефона пользователя | varchar(17) |
| password | Пароль пользователя | varchar(70) |
| DOB | Дата рождения пользователя | date |
| gender | Пол пользователя (0-мужской, 1-женский) | tinyint |
| city\_id | Идентификатор города | int |
| organization\_id | Идентификатор организации | int |
| person\_id | Идентификатор персоны | int |

Таблица 5 - Структура таблицы “articles” – добавь номер п/п в секции, кол.стр.

|  |  |  |
| --- | --- | --- |
| Имя столбца | Назначение | Тип |
| article\_id | Идентификатор статьи | auto increment int |
| name | Название статьи | varchar(150) |
| description | Описание статьи | varchar(255) |
| author\_id | Идентификатор пользователя | int |
| status\_id | Идентификатор статуса | int |
| digest\_section\_id | Идентификатор раздела сборника | int |

Таблица 6 - Структура таблицы “status”

|  |  |  |
| --- | --- | --- |
| Имя столбца | Назначение | Тип |
| status\_id | Идентификатор статуса | auto increment int |
| name | Название статуса | varchar(40) |

Таблица 7 - Структура таблицы “articles\_directors”

|  |  |  |
| --- | --- | --- |
| Имя столбца | Назначение | Тип |
| article\_id | Идентификатор статьи | int |
| director\_id | Идентификатор научного руководителя | int |

Таблица 8 - Структура таблицы “articles\_collaborators” – добавь номер в списке

|  |  |  |
| --- | --- | --- |
| Имя столбца | Назначение | Тип |
| article\_id | Идентификатор статьи | int |
| collaborator\_id | Идентификатор соавтора | int |

Таблица 9 - Структура таблицы “payments”

|  |  |  |
| --- | --- | --- |
| Имя столбца | Назначение | Тип |
| payment\_id | Идентификатор платежа | int |
| amount | Идентификатор соавтора | int |
| date | Дата поступления платежа | date |
| type\_id | Идентификатор типа платежа | int |
| article\_id | Идентификатор статьи | int |

Таблица 10 - Структура таблицы “payment\_types”

|  |  |  |
| --- | --- | --- |
| Имя столбца | Назначение | Тип |
| type\_id | Идентификатор типа платежа | int |
| name | Название способа, которым отправлены средства | varchar(30) |

Таблица 11 - Структура таблицы “persons\_payments”

|  |  |  |
| --- | --- | --- |
| Имя столбца | Назначение | Тип |
| person\_id | Идентификатор персоны | int |
| payment\_id | Идентификатор платежа | int |

Таблица 12 - Структура таблицы “ organizations\_payments ”

|  |  |  |
| --- | --- | --- |
| Имя столбца | Назначение | Тип |
| organization\_id | Идентификатор организации | int |
| payment\_id | Идентификатор платежа | int |

Таблица 13 - Структура таблицы “digests” – год зачем? Номер тома надо

|  |  |  |
| --- | --- | --- |
| Имя столбца | Назначение | Тип |
| digest\_id | Идентификатор сборника | auto increment int |
| publication\_year | Год публикации сборника | int |
| conference\_id | Идентификатор конференции | int |

Таблица 14 - Структура таблицы “sections”

|  |  |  |
| --- | --- | --- |
| Имя столбца | Назначение | Тип |
| section\_id | Идентификатор раздела | auto increment int |
| name | Название раздела | varchar(100) |

Таблица 15 - Структура таблицы “digests\_sections”

|  |  |  |
| --- | --- | --- |
| Имя столбца | Назначение | Тип |
| digest\_section\_id | Идентификатор раздела сборника | auto increment int |
| section\_num | Номер раздела в сборнике | int |
| section\_id | Идентификатор раздела | int |
| digest\_id | Идентификатор сборника | int |

Таблица 16 - Структура таблицы “conferences”

|  |  |  |
| --- | --- | --- |
| Имя столбца | Назначение | Тип |
| conference\_id | Идентификатор конференции | auto increment int |
| name | Название конференции | varchar(150) |
| registration\_date | Дата окончания регистрации | datetime |
| start\_date | Дата начала проведения конференции | date |
| end\_date | Дата окончания проведения конференции | date |
| price | Размер организационного взноса | int |
| organization\_id | Идентификатор организации, проводящей конференцию | int |
| city\_id | Идентификатор города, в котором проводится конференция | int |

Таблица 17 - Структура таблицы “notices”

|  |  |  |
| --- | --- | --- |
| Имя столбца | Назначение | Тип |
| id | Идентификатор объявления | auto increment int |
| text | Текст объявления | text |
| date | Дата публикации объявления | date |
| conference\_id | Идентификатор конференции | int |

### **2.3.3 Описание связей**

У одного пользователя может быть одна или много публикаций. А у каждой статьи только один автор - абсурд. У каждой сущности присутствует собственный идентификатор. Значит присутствует не идентифицирующая связь один-ко-многим между сущностями «users» и «articles».

Каждый пользователь может проживать только в одном городе, однако каждый город может содержать много пользователей, или не содержать их вовсе. У каждой сущности присутствует собственный идентификатор. Значит присутствует не идентифицирующая связь один-ко-многим между сущностями «cities» и «users».

Каждый пользователь может быть закреплен за одной организацией, но к любой организации может быть прикреплено много пользователей. У каждой сущности присутствует собственный идентификатор. Значит присутствует не идентифицирующая связь один-ко-многим между сущностями «organization» и «users».

Каждый пользователь является персоной и у каждой из двух сущностей имеются собственные идентификаторы. Именно из-за этого связь между таблицами «persons» и «users» не идентифицирующая один-к-одному

Одна статья должна иметь только один статус. У каждой сущности присутствует собственный идентификатор. Поэтому связь между сущностями «articles» и «status» не идентифицирующая один-к-одному.

Также одна статья может иметь много научных руководителей, или не иметь их вовсе. У каждой сущности присутствует собственный идентификатор. Следовательно, между “articles” и “persons” присутствует не идентифицирующая связь многие-ко-многим. Однако мы не можем установить прямую связь многие-ко-многим, между двумя таблицами и исправляем это посредством вспомогательной промежуточной таблицы “articles\_directors” [2]. После этого мы получим две связи один-ко-многим.

Точно также одна статья может иметь много соавторов, или не иметь их вовсе. У каждой сущности присутствует собственный идентификатор. Следовательно, между “articles” и “persons” присутствует не идентифицирующая связь многие-ко-многим. Однако, как уже было сказано, мы не можем установить прямую связь многие-ко-многим, между двумя таблицами и исправляем это посредством вспомогательной промежуточной таблицы “articles\_collaborators” [2]. После этого мы получим две связи один-ко-многим.

Любая конференция проводится в одном городе, а в сущностях «cities» и «conferences» имеются собственные идентификаторы. Именно поэтому связь между ними является не идентифицирующей один-к-одному.

Также любая конференция проводится лишь в одной организации. Сущности «organizations» и «conferences» имеются собственные идентификаторы. Именно поэтому связь между ними является не идентифицирующей один-к-одному.

Одна конференция может иметь один или много сборников. У каждой сущности присутствует собственный идентификатор. Следовательно, между сущностями «conferences» и «digests» присутствует не идентифицирующая связь один-ко-многим.

Иногда по касательно каких-либо конференций необходимо публиковать объявления. Одна конференция может содержать много объявлений, или не содержать их вовсе. У каждой сущности присутствует собственный идентификатор. Исходя из этого, между сущностями «conferences» и «notices» присутствует не идентифицирующая связь один-ко-многим.

Предполагается, что в разных сборниках могут содержаться одни и те же разделы, и для того, чтобы это реализовать добавляется вспомогательная промежуточная таблица “digests\_sections”. К ней не идентифицирующей связью один-ко-многим присоединены две сущности “digests” и “sections”.

Каждая статья может быть опубликована всего один раз, а у сущностей “articles” и “digests\_sections” присутствуют собственные идентификаторы. Именно из-за этого связь между данным таблицами не идентифицирующая один-к-одному.

У каждой опубликованной статьи должен быть оплачен организационный взнос. Он может быть оплачен одним платежом, но в случае каких-либо обстоятельств платеж можно разбить и на 2 части. Обе сущности содержат уникальные идентификаторы. Из этого можно сделать вывод, что связь между таблицами “articles” и “payments” не идентифицирующая связь один-ко-многим.

Один платеж должен оплачиваться только одним способом. У каждой сущности присутствует собственный идентификатор. Поэтому связь между сущностями «payments» и «payment\_types» не идентифицирующая один-к-одному.

Много различных персон могут внести много платежей, либо платеж может быть внесен не персоной вовсе. У каждой сущности присутствует собственный идентификатор. Следовательно, между “payments” и “persons” присутствует не идентифицирующая связь многие-ко-многим. Однако мы не можем установить прямую связь многие-ко-многим, между двумя таблицами и исправляем это посредством вспомогательной промежуточной таблицы “persons\_payments” [2]. Также один платеж может поступить только от одной персоны, а следовательно, после этого мы получим две связи один-к-одному.

Точно также много различных организаций могут внести много платежей, либо платеж может быть внесен не организацией вовсе. У каждой сущности присутствует собственный идентификатор. Следовательно, между “articles” и “organizations” присутствует не идентифицирующая связь многие-ко-многим. Однако, как уже было сказано, мы не можем установить прямую связь многие-ко-многим, между двумя таблицами и исправляем это посредством вспомогательной промежуточной таблицы “organizations\_payments” [2]. Также один платеж может поступить только от одной организации, а следовательно, после этого мы получим две связи один-к-одному.

3 Программная реализация ресурса для проведения научных конференций

## **3.1 Структура web-приложения**

Тут диаграмма того, где что хранится, а после описание этой диаграммы

## **3.2 Описание методов REST api**

Здесь огромная таблица, содержащая информацию об api, а также стоит добавить о том, что защита данных

## **3.3 Тестирование разработанного продукта**

Ну тут тупо скрины и показывать процесс что происходит когда тыкаешь куда то

Вывод

Тут должен быть вывод

Список использованных источников – отступы!!!

1. Академик [Электронный ресурс] // dic.academic.ru. Режим доступа: https://dic.academic.ru/dic.nsf/ruwiki/288545 свободный. – Загл. с экрана. – (дата обращения 06.05.2021)
2. Гольцман Виктор. MySQL 5.0 / В. Гольцман – Санкт-Петербург: Бином, 2011. – 321 с.
3. Давыдова, Е.М. Базы данных. / Е.М. Давыдова, Н.А. Новгородова. — Электрон. дан. — М.: ТУСУР, 2007. — 166 с. — Режим доступа: http://e.lanbook.com/book/11636 — Загл. с экрана.
4. Кузнецов М.В., Симдянов И.В. - MySQL 5 - 2010 – 287с.
5. Международная научно-практическая конференция студентов, аспирантов и молодых ученых “МОЛОДЕЖЬ И СОВРЕМЕННЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ” [Электронный ресурс] // msit.tpu.ru. Режим доступа: http://msit.tpu.ru/ свободный. – Загл. с экрана. – (дата обращения 06.05.2021)
6. Международная научно-техническая конференция “Измерение, контроль, информатизация” [Электронный ресурс] // mca.altstu.ru. Режим доступа: http://mca.altstu.ru/ свободный. – Загл. с экрана. – (дата обращения 06.05.2021)
7. Миковски, М. С. Разработка одностраничных приложений. / М. С. Миковски, Д. К. Пауэлл. - М.: Пресс, 2014. - 512 с.
8. Морозов Е. А. Анализ предметной области и концептуальное проектирование базы данных. – М: МИСиС, 2018. – 44 с.
9. Научная сеть мероприятий и форумов для ученых и интеллектуальной молодежи ЛОМОНОСОВ [Электронный ресурс] // lomonosov-msu.ru. Режим доступа: https://lomonosov-msu.ru/ свободный. – Загл. с экрана. – (дата обращения 06.05.2021)
10. Официальная документация по React. [Электронный ресурс]. Режим доступа: https://ru.reactjs.org/ (дата обращения: 06.05.2021).
11. Способы защиты информации [Электронный ресурс] // searchinform.ru. Режим доступа: https://searchinform.ru/informatsionnaya-bezopasnost/zaschita-informatsii/sposoby-zaschity-informatsii/ свободный. – Загл. с экрана. – (дата обращения 06.05.2021)
12. Эспозито, Д. Разработка современных веб-приложений. Анализ предметных областей и технологий. / Дино Эспозито - М.: Вильямс, 2017. - 464 с.
13. Digital 2020: global digital overview [Электронный ресурс] // datareportal.com. Режим доступа: https://datareportal.com/reports/digital-2020-global-digital-overview свободный. – Загл. с экрана. – Текст на экране англ. – (дата обращения 06.05.2021)
14. MySQL [Электронный ресурс] // mysql.ru. Режим доступа: http://www.mysql.ru/ свободный. – Загл. с экрана. – (дата обращения 06.05.2021)
15. Visual Studio Code [Электронный ресурс] // code.visualstudio.com. Режим доступа: https://code.visualstudio.com/ свободный. – Загл. с экрана. – (дата обращения 06.05.2021)