

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА №51

КУРСОВАЯ РАБОТА
ЗАЩИЩЕНА С ОЦЕНКОЙ
РУКОВОДИТЕЛЬ

доцент, ст. преподаватель
должность, уч. степень, звание

подпись, дата

Е.Д. Пойманова
инициалы, фамилия

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К КУРСОВОЙ РАБОТЕ

ИНФОРМАЦИОННАЯ СИСТЕМА УЧЕТА ПУБЛИКАЦИОННОЙ
АКТИВНОСТИ СТУДЕНТОВ

по курсу: КОМПЬЮТЕРНОЕ ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. № 5822

подпись, дата

Е.Д. Энс
инициалы, фамилия

Санкт-Петербург 2022

Содержание

1. Цель работы	3
2. Описание предметной области	3
3. Основные средства	3
4. Описание структуры проекта	5
5. Ход реализации	7
6. Выводы	7
Список используемой литературы	9

1. Цель работы

Целью данной курсовой работы является обзор дипломной работы, описание предметной области, составление структуры проекта, выбор средств проектирования, а также описание хода реализации.

2. Описание предметной области

В связи с необходимостью учета публикационной активности студентов (с целью отчетов, поощрений научно-исследовательской деятельности студентов в виде повышенных стипендий и т.д.) требуется создание информационной системы, которая будет содержать соответствующие сведения, для автоматизации процесса учета активности.

Система должна хранить информацию о всех загруженных публикациях, их характеристик; распределении студентов по группам; персональную информацию о каждом студенте, список всех его публикаций.

Такая система имеет следующие преимущества:

- снижение трудоемкости сотрудников кафедры;
- сокращение времени обработки и получения данных;
- формирование индивидуального списка всех работ студента;
- повышение степени защищенности информации;

Данная информационная система создается для того, чтобы сотрудник кафедры мог беспрепятственно обратиться к любой публикации какого-либо студента, ознакомиться с ней, либо получить список всех публикаций определенного студента.

3. Основные средства

Для реализации данной системы существуют различные инструменты, но воспользуемся инструментами, которые входят в стек PERL. В реализации данной системы будет два этапа разработки: Back-End и Front-End.

3.1. Серверная часть – Back-End

Для написания серверной части существует множество фреймворков, к примеру, AdonisJs, Napi, Express, Meteor, Sails, Koa, LoopBack, Derby. В данной работе был выбран фреймворк Express в связке с NodeJs.

Для работы с базами данных потребуется система управления. К таким системам относятся: Oracle Database, Firebird, Interbase, Informix, MS SQL Server, PostgreSQL, MySQL. Но для разработки информационной системы был выбран PostgreSQL, так как у него множество возможностей. Он поддерживает сложные структуры и широкий спектр встроенных и определяемых пользователем типов данных. Также он обеспечивает расширенную ёмкость данных и заслужил доверие бережным отношением к целостности данных. Даже если для разработки системы все продвинутые функции хранения данных могут не потребоваться, то в будущем потребности могут возрасти, и есть несомненное преимущество в том, чтобы иметь все эти функции это под рукой.

Также для того, чтобы связывать программный код с базами данных необходимо использовать ORM. Примерами таких технологий являются Bookshelf, Waterline, Objection.js, Sequelize, Mongoose. Каждая из них обладает своими особенностями, но в данной работе была выбрана именно Sequelize, так как она является одной из наиболее популярных, продвинутых и “проверенных временем” технологий.

3.2. Клиентская часть – Front-End

При разработке клиентской части будет использоваться следующий стек технологий:

1. React JS - JavaScript-библиотека для создания пользовательских интерфейсов
2. React Bootstrap. Будет использоваться для того, чтобы облегчить процесс вёрстки
3. Axios - JavaScript-библиотека для запросов к серверу
4. React-router-dom. Будет использоваться для навигации по странице
5. MobX - это автономная библиотека, для управления фронтенд-состоянием приложения.

4. Описание структуры проекта

На рисунке 1 изображен процесс работы веб-приложения. После того как пользователь открыл веб-приложение или нажал какую-то кнопку должно произойти какое-то действие, и он должен получить данные с сервера. Сервер расположен по определенному адресу (например, <http://server.com>), по этому адресу отправляется запрос. В свою очередь сервер взаимодействует с базой данных, как-то обрабатывает данные, происходят какие-то процессы, после чего сервер возвращает эти данные на клиент, и пользователь уже получает их в виде какой-то определенной информации.

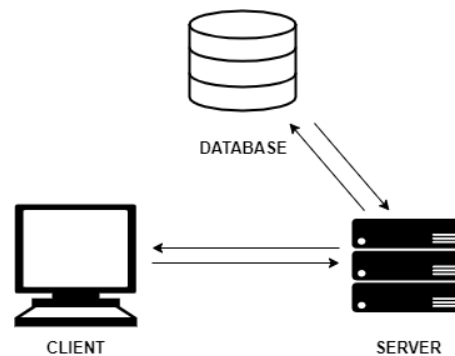


Рисунок 1 - Процесс работы веб-приложения

В ходе разработки серверной части будет построена диаграмма базы данных, которая будет состоять из 4 таблиц (Рисунок 2).

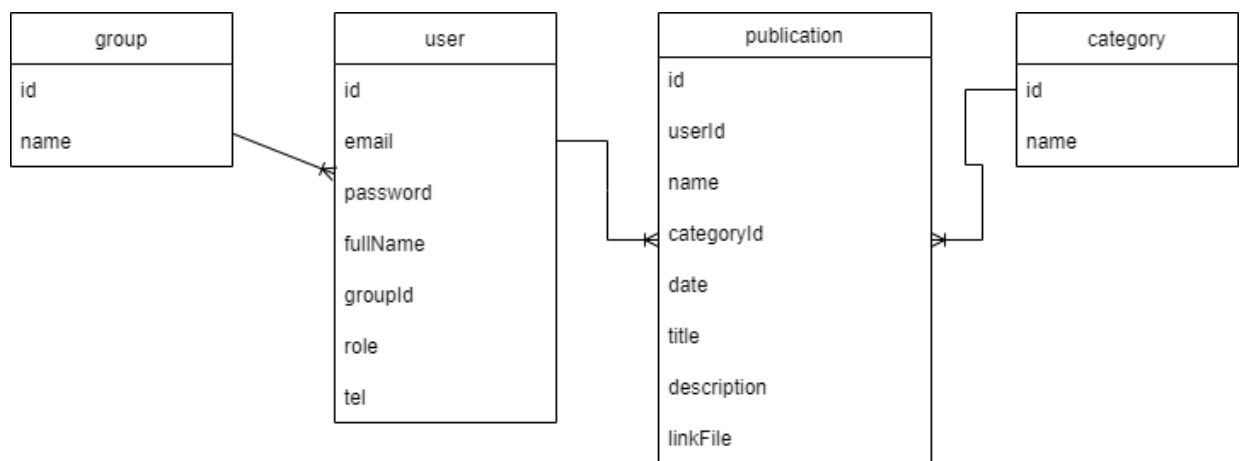


Рисунок 2 – Диаграмма схемы базы данных

Поля таблицы user:

- *id* – уникальный идентификатор пользователя;
- *email* – логин, по которому будет осуществляться вход в систему;
- *password* – пароль пользователя;
- *fullName* – поле, которое содержит ФИО;
- *groupId* – содержит id группы, к которой принадлежит пользователь (студент)
- *role* – роль пользователя;
- *tel* – поле с контактным номером телефона пользователя;

Поля таблицы group:

- *id* – уникальный идентификатор группы;
- *name* – поле, в котором будет храниться номер группы;

Поля таблицы publication:

- *id* – уникальный идентификатор публикации;
- *userId* - уникальный идентификатор, который показывает какому пользователю принадлежит публикация;
- *name* – название публикации;
- *categoryId* - уникальный идентификатор, который указывает к какой категории относится публикация;
- *date* – дата публикации;
- *title* – поле, которое будет содержать заголовок для какой-либо дополнительной информации;
- *description* - поле, которое будет содержать описание какой-либо дополнительной информации;
- *linkFile* – поле, которое содержит ссылку для скачивания электронной версии публикации;

Поля таблицы category:

- *id* – уникальный идентификатор категории;
- *name* – поле, в котором будет храниться название категории;

В результате разработки будет реализовано полноценное веб-приложение, с помощью которого студент сможет зарегистрироваться (в дальнейшем выполнять вход в личный кабинет), выгружать свои публикации, при этом указывая определенные параметры, а также просматривать список со своими работами. Работник кафедры в свою очередь также сможет авторизовываться, предварительно зарегистрировавшись, но при этом иметь права администратора, с помощью которой он сможет выполнять поиск публикаций по определенным параметрам, просматривать список всех публикаций любого студента, взаимодействовать с базой данных, путем добавления/удаления новых групп, категорий публикаций, а также добавлять новых администраторов.

5. Ход реализации

Начальным этапом для реализации данной системы является создание, настройка и запуск сервера, затем необходимо настроить подключение к базе данных. В ходе разработки серверной части будет построена диаграмма базы данных.

Далее эту диаграмму необходимо перенести в проект. Затем будет реализована схема того, как эти данные будут храниться в базе данных; в проекте будут настроены модели, их поля и связи.

Следующим этапом будет реализация полноценной регистрации и авторизации пользователей по JSON Web Token (JWT). Также будет реализован обработчик ошибок при авторизации.

Второй стадией разработки веб-приложения является Front-End разработка, в которой будет описано структура всего приложения; будет написан пользовательский интерфейс; также будет реализована навигация и правильное отображение необходимых страниц; взаимодействие с сервером.

6. Выводы

Таким образом, в данной курсовой работе была рассмотрена предметная область проекта, проанализированы основные инструменты.

Также была составлена диаграмма схемы базы данных, выделены основные стадии автоматизации. Приведены теоретические сведения о каждой стадии по отдельности, выбраны средства для их реализации и описан примерный ход реализации

Список используемой литературы

1. PERN Stack. URL: <https://docs.gitlab.com/ee/ci/introduction/> . (дата обращения 16.04.2022)
2. PostgreSQL Documentation. URL: <https://www.postgresql.org/docs/current/> . (дата обращения 16.04.2022)
3. Express Documentation. URL: <https://devdocs.io/express/> . (дата обращения 16.04.2022)
4. React Overview. URL: <https://en.reactjs.org/docs/react-api.html> . (дата обращения 16.04.2022)
5. NodeJS docs. URL: <https://nodejs.org/api/> . (дата обращения 16.04.2022)