МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»

Кафедра информатики и программирования

Разработка Web-платформы для дистанционного обучения

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 441 группы
направления 02.03.03 Математическое обеспечение и администрирование
информационных систем
факультета компьютерных наук и информационных технологий
Бурдавицына Артёма Андреевича

Научный руководитель,
доцент кафедры ИиП,
к.э.н. _____ Л.В.Кабанова
подпись, дата

Зав. кафедрой ИиП,
к.ф.-м.н., доцент _____ М.В. Огнева
подпись, дата

Саратов 2022

ВВЕДЕНИЕ

Актуальность темы. С каждым годом растут темпы всеобщей компьютеризации, компьютеры выполняют всё большие и большие задачи. Также постоянно возникает потребность в удаленной работе, удаленном обучении. В связи с этим увеличивается потребность в системах, позволяющих реализовать подобное взаимодействие пользователей.

Одна из важнейших задач – обеспечение удаленного взаимодействия студентов и преподавателей в университете с помощью онлайн-платформы. Эта проблема особенно актуальна в рамках дистанционной формы обучения, в рамках которой невозможно личное общение преподавателя и студента.

Цель бакалаврской работы – реализация web-платформы, позволяющей дистанционно осуществлять взаимодействие студента и преподавателя в рамках учебного процесса.

Поставленная цель определила **следующие задачи**:

1) Выбрать технологию, используемую для реализации проекта, изучить необходимые фреймворки.
2) Разработать и настроить систему аккаунтов и контроля доступа.
3) Изучить специфику расписания, настроить его автоматизированную синхронизацию с сайтом СГУ.
4) Реализовать систему взаимодействия студентов и преподавателей – добавление домашних заданий, ответов на них и получение отчетности преподавателем.
5) Реализовать систему, позволяющую коммуницировать преподавателю со студентами, а также студентам и преподавателям между собой.
6) Реализовать веб-интерфейс и мобильное приложение для данной платформы.

Методологические основы разработки Web-платформы для дистанционного обучения представлены в работах П.Сэмми, Л.Арно, С.П.Хабарова, М.Л.Шилкиной, Грофф Джеймс Р., Вайнберг Пол Н., Оппель Эндрю Дж.

Практическая значимость бакалаврской работы состоит в реализации платформы, позволяющей осуществлять взаимодействие студента и преподавателя. В возможности платформы входит онлайн синхронизация расписания с сайтом СГУ, работа с домашними заданиями, разделение пользователей по ролям, чат и мобильное приложение.

Структура и объём работы. Бакалаврская работа состоит из введения, 2 разделов, заключения, списка использованных источников и 2 приложений. Общий объем работы – 61 страница, из них 49 страниц – основное содержание, включая 36 рисунков, список использованных источников информации – 20 наименований.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Первый раздел «Архитектура web-приложения» посвящен обзору методов и технологий, использующихся для реализации web-приложений, а также постановке задач, обзору существующих решений, приведен обзор инструментальных средств, необходимых для реализации цели и задач работы.

1. Архитектура web-приложения

Web-приложения представляют собой особый тип ПО, построенного по архитектуре "клиент-сервер". Её особенность заключается в том, что само приложение находится и выполняется на сервере - клиент при этом получает только результаты работы [1].

На стороне сервера для реализации бизнес-логики могут использоваться следующие технологии:

- PHP (Laravel, CodeIgniter и т.д)
- Node.js (Express и т.д)
- Python (Django и т.д)
- Ruby (Ruby on Rails и т.д)
- Java (Spring и т.д)
- C# (ASP.NET и т.д)

Нас стороне клиента для реализации интерфейса могут использоваться следующие технологии:

- HTML
- CSS, SCSS, SASS и т.д
- JavaScript и его фреймворки jQuery, Angular, React, Vue.js и т.д

1.1 Клиент-серверная архитектура приложения

«Клиент-сервер» (client-server) – вычислительная архитектура, задачи в которой распределены между поставщиками услуг – серверами и заказчиками услуг – клиентами.

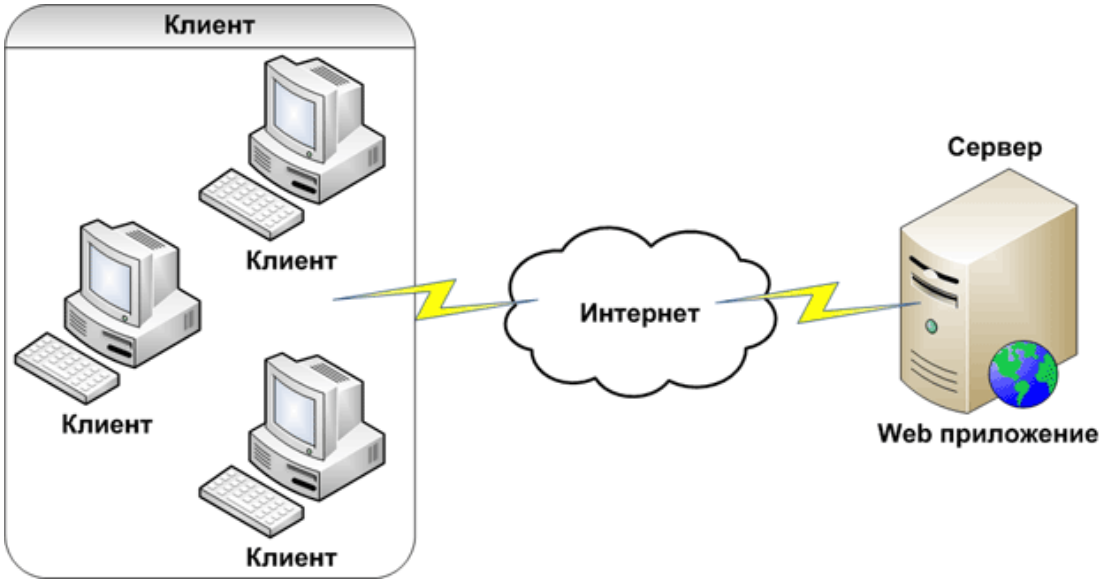

Рис.1 – Схема работы двухуровневого клиент-серверного приложения

Принцип работы клиент-серверной архитектуры:

1) Клиент формирует и посылает запрос на сервер

2) Сервер производит необходимые манипуляции с данными, формирует результат и передаёт его клиенту

3) Клиент получает результат, отображает его на устройстве вывода и ждет дальнейших действий пользователя

4) Цикл повторяется, пока пользователь не закончит работу с сервером.

1.2 MVC

MVC (Model-View-Controller) – разделение приложения на логику и пользовательский интерфейс. Данный паттерн используется для удобства независимой модификации модулей приложения.

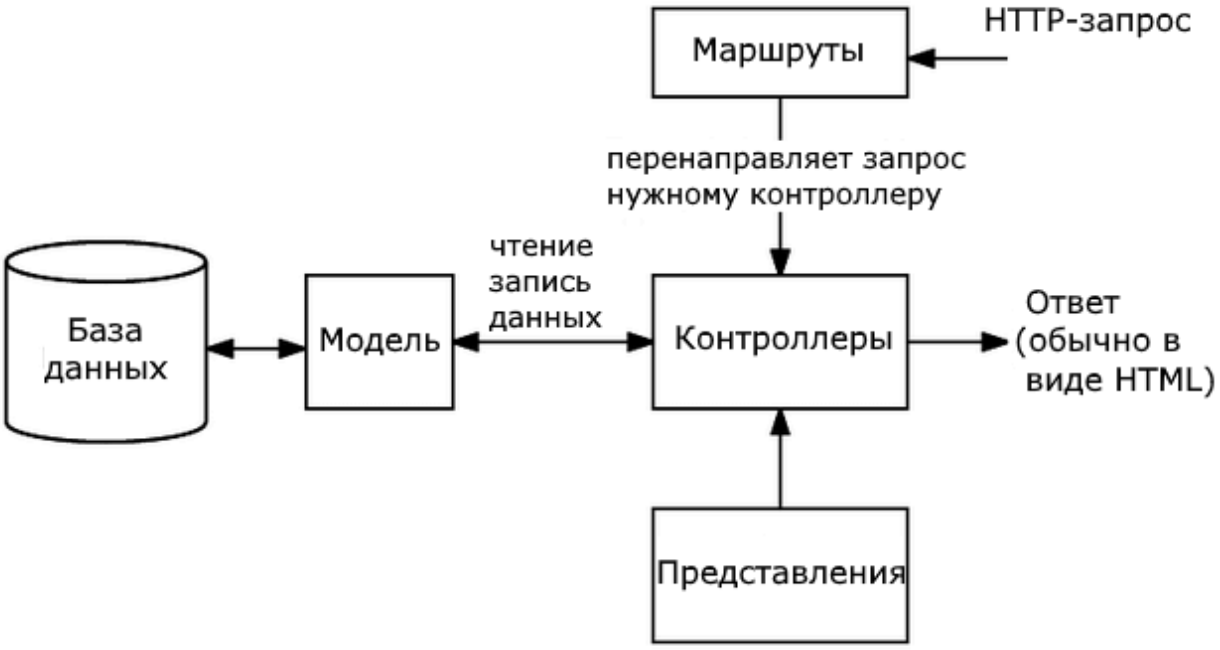

Рисунок 2 – Схема работы паттерна MVC

Модель (model) осуществляет работу с данными приложения, реагирует на команды от контроллера. Представление (view) отвечает за формирование интерфейса пользователя, представляет данные, полученные из Модели пользователю. Контроллер (controller) интерпретирует действия пользователя с помощью интерфейса, осуществляет работу с моделями, отвечает за связь Модели и Представления, направляет данные в обе стороны. Основной принцип MVC состоит в разделении. Бизнес-логика (Модель) отделяются от визуализации (Представление, Вид). Это позволяет использовать части кода повторно, что улучшает восприятие проекта, уменьшает время на разработку, доработку и поддержку [5].

1.3 Rest API

REST (Representational State Transfer) – передача состояния представления – архитектурный стиль взаимодействия компонентов в сети. Для приложений, построенных с помощью REST используется термин RESTful. API (Application Programming Interface) – набор методов, с помощью которых могут взаимодействовать компоненты в сети.

1.4 Базы данных и системы управления базами данных

База данных – совокупность данных, хранимых в определенной иерархичной системе, манипулирование с которыми осуществляется в соответствии с правилами моделирования данных.

Система управления базами данных (СУБД) – программное обеспечение, обеспечивающее создание, использование и администрирование баз данных. Данное средство позволяет создавать базы данных и работать с их содержимым. СУБД обеспечивает безопасность, надежность и целостность хранения данных, предоставляет средства для администрирования.

1.5 JSON Web Token

JSON Web Token (JWT) – открытый стандарт создания токенов доступа, использующий формат JSON. Используется для аутентификации в клиент-серверных приложениях. Токен создается на стороне сервера с использованием секретной последовательности символов и данных клиента. Далее токен передается клиенту для дальнейшего использования.

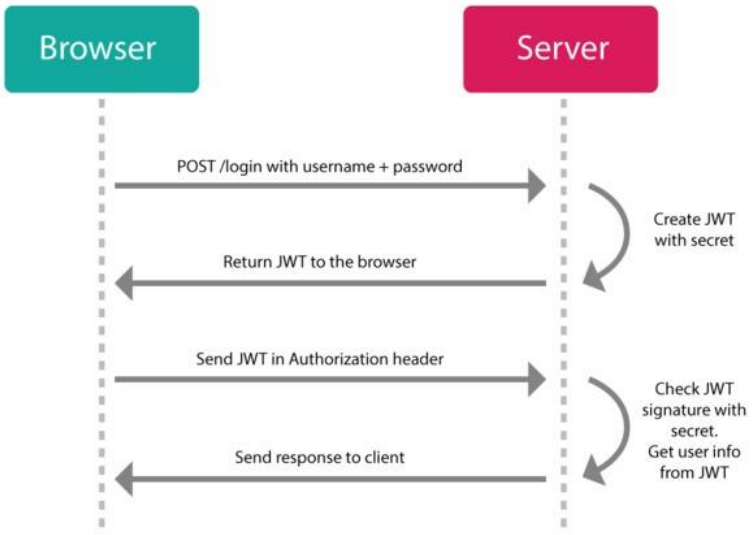

Рисунок 3 – Схема авторизации с использованием JWT

1.6 PWA

PWA (Progressive Web App – прогрессивное веб приложение) – технология, позволяющая трансформировать сайт в мобильное приложение [13]. PWA является гибким решением, позволяющим быстро создать мобильное приложение для iOS и Android. Технология позволяет максимально имитировать работу реального мобильного приложения:

- Отправка push-уведомлений

- Офлайн режим работы
- Доступ к ПО устройства
- Ярлык приложения для рабочего стола

Второй раздел «Реализация приложения» посвящен реализации web-приложения, всех его функций и мобильного приложения.

2.1 Используемый стек технологий, архитектура проекта

В данной работе ролью клиента будет являться веб-браузер пользователя, установленный на его ПК. Сервером будет являться удаленный компьютер под управлением ОС Centos 7 с установленным дистрибутивом Node.js версии 14.3.0.

2.1.1 Back-end

В качестве языка программирования выбран Node.js Его выбор был обусловлен широкой поддержкой и большим количеством дополнений. С помощью фреймворка Express.js было реализовано web-приложение и API.

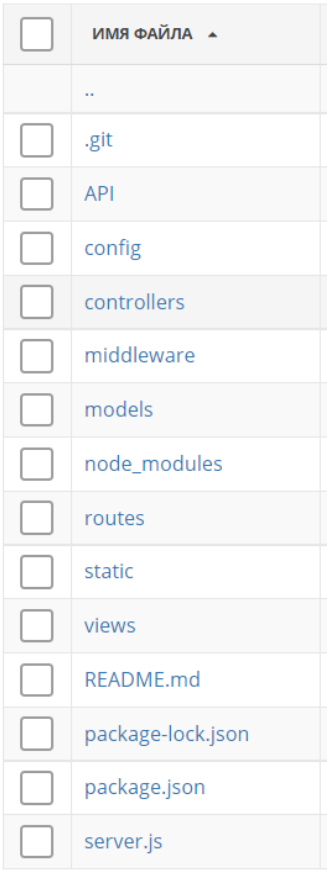

Рисунок 4 – структура проекта

2.1.2 База данных

В качестве базы данных будем использовать MySQL. Выбор был обусловлен широкой поддержкой и удобством администрирования.

2.1.3 Front-end

Для создания пользовательского интерфейса используются следующие технологии:

- HTML (HyperText Markup Language) – язык разметки.
- CSS (Cascading Style Sheets) – язык стилей.
- JavaScript – язык программирования.
- Bootstrap – библиотека готовых HTML и CSS шаблонов.
- jQuery – JavaScript-библиотека, содержащая набор готовых функций.

2.2 Регистрация и авторизация пользователей

Для использования данной информационной системы необходимо быть зарегистрированным в ней под соответствующей ролью – студент или преподаватель – и быть авторизованным в своем аккаунте с помощью формы авторизации на сайте. Важно отметить, что пароли от аккаунтов пользователей небезопасно хранить в открытом виде в базе данных, поэтому они шифруются с помощью модуля bCrypt, реализующий одноименную криптографическую хеш-функцию [11].

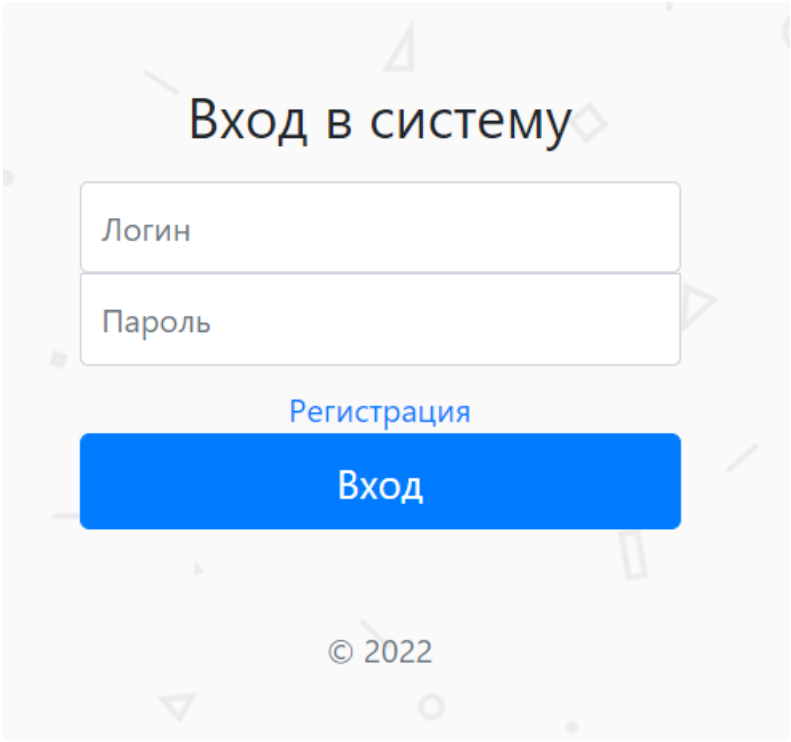

Рисунок 5 – Форма авторизации на сайте

Авторизация происходит с помощью API – данные в полях формы сохраняются и отправляются на сервер с помощью AJAX.

2.3 Контроль доступа

Контроль доступа к содержимому сервиса, контроль ролей пользователей осуществляется на сервере с помощью JSON Web Token – единого стандарта токенов доступа, основанного на формате JSON. JSON Web Token при регистрации или авторизации пользователя на основе его логина и jwtSecret – секретного ключа, хранящегося в config\constants.js. Далее Токен передается в cookies – фрагмент данных, хранящийся на стороне пользователя. При каждом запросе к контенту сайта или обращении к API (в тех местах, где

стоит сделать его недоступным для неавторизованных пользователей) данный токен будет проверяться с помощью middleware\auth.js

2.4 Импорт расписания с сайта СГУ

Сайт СГУ не имеет открытого API для получения расписания, воспользуемся парсингом HTML страницы с помощью библиотеки node-html-parser. Эта библиотека по HTML коду страницы стоит её DOM модель и реализует базовые query запросы к ней. Для получения HTML контента страницы воспользуемся библиотекой request, позволяющей выполнять HTTP и HTTPS запросы.

2.5 Взаимодействие пользователей

В системе предусмотрены следующие взаимодействия между студентами и преподавателями:

- Добавление и редактирование домашних заданий
- Добавление ответов на домашние задание
- Отчет о выполнении домашних заданий
- Возможность оценивания работы преподавателем и оставление комментария

2.5.1 Добавление и редактирование домашних заданий

Преподаватель имеет возможность добавлять домашнее задание для студента на определённую пару.

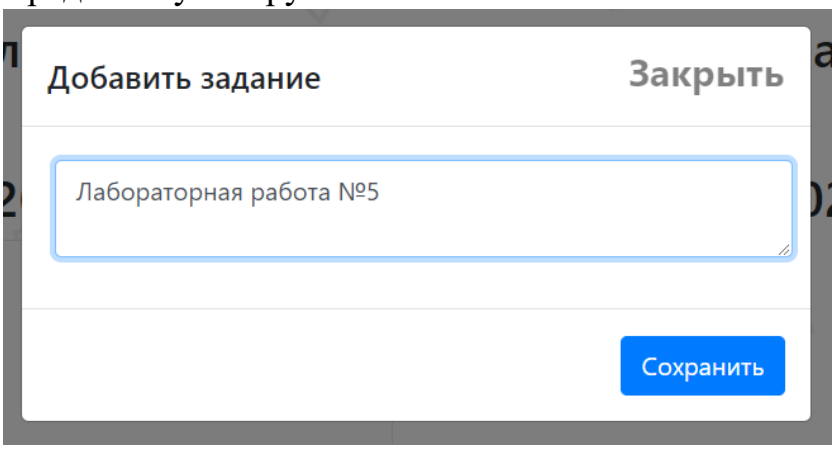

Рисунок 6 – Пример добавления домашнего задания

После подтверждения задание отобразится у студента и у преподавателя.

2.5.2 Добавление ответов на домашние задания

Студент имеет возможность добавить ответ на домашнее задание в виде текста, который также может являться ссылкой. После добавления ответа его можно будет отредактировать.

2.5.3 Отчет о выполнении домашних заданий

Преподаватель имеет возможность получить общую сводку по выполнению задания.

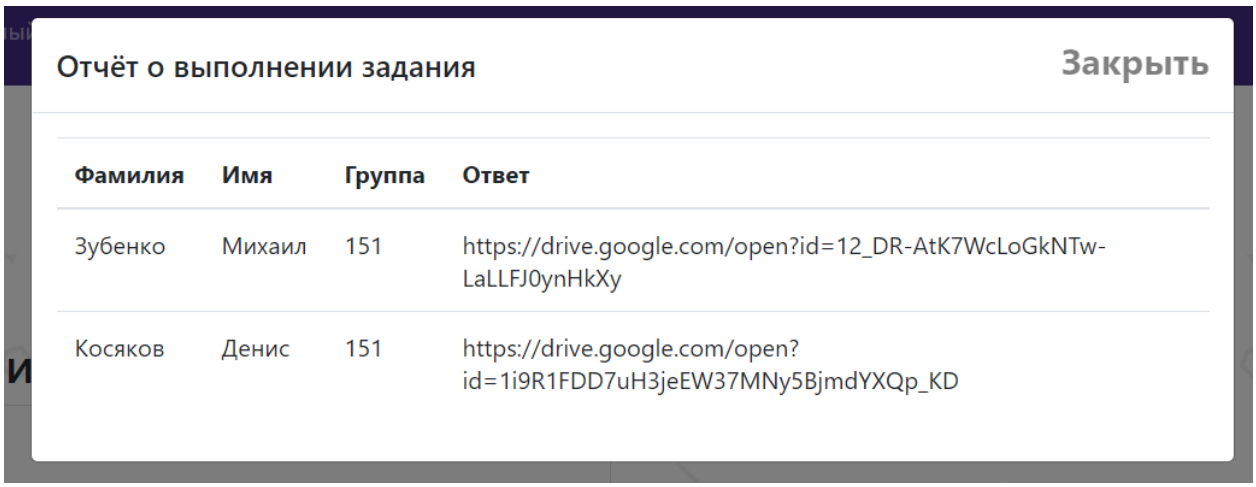

Рисунок 7 – Пример отчета о выполнении задания

2.6 Возможности чата

В разделе Сообщения предусмотрены следующие возможности:

- Обмен текстовыми сообщениями
- Создание личных и групповых чатов

2.6.1 Обмен текстовыми сообщениями

Для полноценной организации работы чата необходимо внедрить функционал обмена сообщениями в реальном времени. Для этой задачи необходимо использовать WebSocket, с помощью него будем отправлять и принимать сообщения.

2.6.2 Создание личных и групповых чатов

Для полноценной реализации системы чата необходимо добавить возможно создавать как личные, так и групповые чаты (рис. 33). Окно выбора собеседников позволяет просматривать всех пользователей системы и выбирать собеседников. Выбор участников диалога будет осуществляться с помощью всплывающего модального окна.

2.7 Progressive Web Application

Для того, чтобы создать PWA приложение, необходимо создать 2 файла: Service Worker и Web App Manifest. Web App Manifest определяет конфигурацию приложения: название, логотипы, ссылку, режим отображения. Service Worker отвечает за работу PWA приложения. В нем добавлены обработчики событий открытия, установки и удаления приложения.

ЗАКЛЮЧЕНИЕ

В условиях постепенного проникновения компьютерной техники во все сферы человеческой деятельности возникает потребность различных образовательных организаций в создании электронных сервисов для взаимодействия студентов и преподавателей. В связи с этим необходимо разрабатывать и поддерживать подобного рода решения для различных процессов в различных сферах человеческой жизни: обучение, работа, развлечения.

В ходе выполнения работы была спроектирована и построена система онлайн взаимодействия студента и преподавателя – электронный дневник студента. Был настроен импорт актуального расписания с сайта СГУ, к нему была привязана система домашних заданий, включившая в себя возможности по выдаче задания, сбора ответов на него, редактирования и сбора отчета о выполнении задания группой.

В стек технологий вошли Node.js и его фреймворк Express, jQuery, AJAX, HTML, CSS, JavaScript, WebSocket и PWA.

Данная система весьма актуальна и может найти своё практическое применение в университете.

Задачи реализованы, цель работы достигнута.

Основные источники информации:

1) П. Сэмми Основы разработки веб-приложений / П. Сэмми – Спб.: ООО «Питер», 2015. – 272 с.: ил
2) Мартин Роберт С. Чистая архитектура. Искусство разработки программного обеспечения / Мартин Роберт С. – Спб.: ООО «Питер», 2018. – 874 с.: ил
3) С. Макконнелл Идеальный код / Макконнелл С. – Спб.: ООО «БХВ-Петербург», 2017. – 896 с.: ил
4) М. Хавербеке Выразительный JavaScript. Современное веб-программирование / М. Хавербеке – Спб.: ООО «Питер», 2019. – 480 с.: ил
5) Express – Node.js web application framework [Электронный ресурс]. URL: https://expressjs.com. Дата обращения – 05.05.2022.
6) Harihara Subramanian, Pethuru Raj Hands-On RESTful API Design Patterns and Best Practices / Harihara Subramanian, Pethuru Raj – En: Packt Publishing, 2019. – 378 с.: ил
7) Л. Арно Проектирование веб-API / Л. Арно – М.: ООО «ДМК Пресс», 2020. – 440 с.: ил
8) С. П. Хабаров, М. Л. Шилкина Построение распределенных систем на базе WebSocket. Учебное пособие для вузов, 2-е изд., стер. / С. П. Хабаров, М. Л. – М.: ООО «Лань», 2022. – 216 с.: ил
9) Гровф Джеймс Р., Вайнберг Пол Н., Оппель Эндрю Дж. SQL. Полное руководство / Гровф Джеймс Р., Вайнберг Пол Н., Оппель Эндрю Дж. . – М.: ООО «Вильямс», 2018. – 960 с.: ил
10) К. Мартин Высоконагруженные приложения. Программирование, масштабирование, поддержка / К. Мартин – Спб.: ООО «Питер», 2022. – 640 с.: ил