

*Васькин Вадим Александрович, магистр, Национальный исследовательский  
Мордовский государственный университет им. Н.П. Огарева*

*Камаева Анастасия Андреевна, магистр, Национальный исследовательский  
Мордовский государственный университет им. Н.П. Огарева*

*Тепяева Юлия Евгеньевна, магистр, Национальный исследовательский  
Мордовский государственный университет им. Н.П. Огарева*

*Ротанов Андрей Сергеевич, магистр, Национальный исследовательский  
Мордовский государственный университет им. Н.П. Огарева*

*Ямашкин Станислав Анатольевич, кандидат технических наук, доцент  
кафедры автоматизированных систем обработки информации и управления,  
Национальный исследовательский Мордовский государственный университет  
им. Н. П. Огарева*

## **РАЗРАБОТКА WEB-ПРИЛОЖЕНИЯ ДЛЯ УПРАВЛЕНИЯ ПРОЕКТАМИ И КОНТРОЛЯ ВЫПОЛНЕНИЯ ЗАДАЧ**

**Аннотация:** В статье рассматривается разработка web-приложения, которое позволяет контролировать выполнение задач сотрудниками предприятия, определять время, затраченное на задания, а также использовать форму обратной связи между ячейками организации, которая позволит исключить необходимость использования сторонних сервисов. Приложение разработано посредством следующих языков программирования: HTML, JS, Python, SQL.

Целью исследования является изучение методов разработки систем с использованием асинхронных запросов, и разработка web-приложения для управления проектами контроля выполнения задач.

**Ключевые слова:** web-приложение, разработка, интерфейс, API, JavaScript, Python, Flask, PostgreSQL.

**Annotation.** The article discusses the development of a web application that allows you to control the execution of tasks by employees of the enterprise, determine the time spent on tasks, and also use the feedback form between the cells of the organization, which will eliminate the need to use third-party services. The application was developed using the following programming languages: HTML, JS, Python, SQL.

The purpose of the study is to study the methods of system development using asynchronous requests and the development of a web application for managing projects to control the execution of tasks.

**Keywords:** web application, development, interface, API, JavaScript, Python, Flask, PostgreSQL.

## **Введение**

В настоящее время абсолютно всем организациям, прямо или косвенно связанным с информационными технологиями, требуется средство управления задачами и проектами. Это набор инструментов, позволяющий прикреплять новые задачи, следить за их выполнением, вести отсчёт времени, потраченный на их них [1].

В связи с этим, компании, у которых нет денег на поддержку программного обеспечения, прибегают к использованию бесплатных, в большинстве случаев недоработанных проектов, или же пользуются полноценными приложениями пробный период.

Существующих решений достаточно много, но, как правило, они не включают в себя средства общения между сотрудниками и руководителем. Вследствие их отсутствия, возникает необходимость прибегать к дополнительным сервисам для обсуждения вопросов, возникающих по ходу выполнения задач.

Цель исследования преследует необходимость реализации следующего ряда задач:

- анализ теоретических аспектов разработки,
- выбор инструментов для разработки, языков программирования,
- разработка серверной и клиентской части приложения.

Современный трекер должен обладать огромным функционалом, позволяющим работать со всевозможными видами задач и подзадач, следить за временем, потраченным на них, включать в себя инструменты для делегирования обязанностей непосредственно внутри приложения, а также иметь средство коммуникации между ячейками организации в виде общего чата по конкретным задачам.

Исходя из вышесказанного, можно сформулировать основные принципы, которым должен соответствовать разрабатываемый трекер задач:

- возможность создания новых дашбордов, задач, назначение пользователей на определённые задачи, реализация замера времени выполнения каждой из задач;

- возможность использования средств обратной связи внутри каждой из задач, с возможностью прикрепления файлов.

В ходе исследования был проанализирован ряд программных продуктов, осуществляющих контроль выполнения задач (Trello, Weeek, Yandex Tracker). Рассмотренные продукты вышеуказанным критериям соответствуют не в полной мере. В результате исследования все вышеуказанные требования в разрабатываемом продукте были учтены.

Реализация всех перечисленных выше условий обязательна для выполнения. Это особенно важно для приложения, которое обеспечивает контроль выполнения задач на предприятии [2].

Для реализации веб-приложения использовались следующие ресурсы: база данных, backend (язык программирования python, фреймворк flask), frontend (языки программирования JavaScript, JQuery, AJAX, WebSocket), в качестве СУБД использовалась PostgreSQL [3].

Flask – фреймворк на языке программирования Python для создания каркасов веб-приложений. AJAX и WebSocket получили широкое применение

в проекте за счет возможности осуществления обращения к серверу без перезагрузки страницы путём осуществления асинхронных запросов в реальном времени [4].

Система предусматривает роли для пользователей: руководитель и сотрудник. Каждый тип наделен особыми правами. У руководителей доступных возможностей будет гораздо больше. Среди них: возможность добавления новой «доски», задач для выполнения, прикрепление (открепление) пользователей на конкретные задачи, наделение пользователей правами. Сотрудники же будут иметь возможность просмотра задач, использовать секундомер для контроля времени их выполнения, а также присоединяться к проектам. Для тех и других реализована возможность открытия мессенджера для обратной связи между типами пользователей, а также прикрепление файлов и вложений.

Серверная и клиентская части приложения взаимодействуют между собой посредством запросов. Например, при заполнении пользователем формы ввода данные упаковываются в REQUEST запрос и передаются на сервер, который в свою очередь перенаправляет их в СУБД. Каждая форма для ввода имеет проверку на корректность введенного текста, которая осуществляется посредством регулярных выражений.

В качестве примера серверной части приложения рассмотрим реализацию функционала регистрации пользователя. После заполнения формы для регистрации, отправляется запрос на сервер, выполняется проверка на корректность введенных данных, проверка на существование электронной почты, хэширования пароля (алгоритм осуществляется посредством модуля hashlib – инструмента для реализации алгоритмов хэширования на языке программирования Python [5]), после чего сервер произведёт запись данных в СУБД, а пользователя перенаправит на страницу авторизации.

Фрагмент программного кода, отвечающий за обработку данных формы регистрации, приведён ниже:

```

if request.method == 'POST':
    name = data["name"]
    email = data["email"]
    password_hash = data["password"]
    user = User.query.filter_by(email=email).first()
    if user:
        flash('Email уже используется')
        return render_template("registr1.html")
    email_correct = Email.query.filter_by(code=code).first()
    if email_correct:
        email=email_correct.email
        new_data = User(name, email, password_hash, created_on, updated_on,
status)

        db.session.add(new_data)
        db.session.commit()
        return redirect('log')

```

Основной задачей в процессе реализации клиентской части приложения было создание коммуникативной среды для взаимодействия между ячейками организации посредством асинхронных запросов. Для отправления и получения сообщений чата приложения в реальном времени было использовано односокетное TCP-соединение между клиентом и сервером, что позволило мгновенно распределять двунаправленные сообщения.

Фрагмент программного кода реализации данной технологии посредством использования в приложении WebSocket, приведён ниже:

```

socket.on( 'connect', function() {
    socket.emit( 'my event', {
        data: 'User Connected'
    } )
var form = $( '#ajax' ).on( 'submit', function( e ) {
    e.preventDefault()
    let user_name = $( 'input.name' ).val()
    let user_input = $( 'textarea.message' ).val()
    var now = new Date().toLocaleString()

```

```

var todo = $( 'input.id_chat' ).val()
socket.emit( 'my event', {
  user_name : user_name,
  message : user_input,
  now : now,
  todo : todo,
  mess : mess
} )
$( 'textarea.message' ).val( '' ).focus()
} )
} )

```

В полях формы ввода пользователем вводится сообщение, сервером считывается уникальный идентификатор пользователя, после чего происходит двунаправленный запрос, выводящий новое сообщение пользователю и отправляющий запрос на сервер для записи в БД.

Разработанное приложение отвечает функциональным требованиям большинства организаций, нуждающихся в контроле выполнения задач, помогая руководителям следить за качеством выполненной сотрудниками работы. Для повышения функциональности существующего программного продукта может быть предложен ряд улучшений:

- Добавление календаря событий для удобной визуализации тайминговых процессов;
- Разработка методики для расчёта коэффициента полезного действия сотрудников на основе получаемых данных [6];
- Генерация отчетной статистики за выбранный руководителем период времени по выполненным сотрудниками задачам в конкретных проектах.

### **Выводы**

В ходе работы было разработано веб-приложение, которое сможет обеспечить контроль над проектами, а также позволит пользователям внутри приложения вести переговоры по поводу той или иной задачи в чате, специально спроектированном под данное приложение. В приложении

применяются современные технологии для обработки асинхронных запросов, а также используется система управления базами данных PostgreSQL для безопасного хранения данных разработанной системы. Приложение представляет собой готовый программный продукт, соответствующий выдвинутым критериям.

Предложенный трекер задач оснащён функционалом, который способен осуществлять контроль за сотрудниками предприятия, адаптирован под любые виды задач, вне зависимости от направления деятельности организации. В отличие от своих аналогов обладает инструментом обратной связи между ячейками организации, что избавит от необходимости прибегать к дополнительному программному обеспечению.

Следующим шагом разработки приложения будет включение в проект функций для предоставления руководителю статистики по выполненным сотрудниками задачам в конкретных проектах. Новые функции приложения должны помочь руководителям выделять наиболее ценных сотрудников.

#### **Библиографический список:**

1. Ерофеев, В. Т. Проблемы повышения качества подготовки инженерных кадров и пути их решения / В. Т. Ерофеев, В. В. Афонин // Инновационное направление учебно-методической и научной деятельности кафедр материаловедения и технологий конструкционных материалов: материалы Всероссийского совещания с международным участием зав. кафедрами материаловедения и технологий конструкционных материалов, Саранск, 24–29 октября 2016 года. – Саранск: Национальный исследовательский Мордовский государственный университет им. Н.П. Огарёва, 2016. – С. 16-19. – EDN YMLVRJ.
2. Демина А. В., Мищенко А. Ю. Разработка приложения" менеджер проектов" / А. В. Демина, А. Ю. Мищенко //Вестник Саратовского государственного социально-экономического университета. – 2020. – №. 2 (81). – С. 133-137. Текст: непосредственный.

3. Flask-SQLAlchemy documentation: сайт. – URL: <https://flask-sqlalchemy.palletsprojects.com/en/2.x/> (дата обращения 16.05.2021). – Режим доступа: свободный. – Текст: электронный.
4. Введение в AJAX: сайт. – URL: <https://developer.mozilla.org/ru/docs/> (дата обращения 16.05.2021). – Режим доступа: свободный. – Текст: электронный.
5. Добавление аутентификации в ваше приложение с помощью Flask-Login: сайт. – URL: <https://www.digitalocean.com/community/tutorials/how-to-add-authentication-to-your-app-with-flask-login-ru> (дата обращения 16.05.2021). – Режим доступа: свободный. – Текст: электронный.
6. Ерофеев, В. Т. Методы обработки экспериментальных данных / В. Т. Ерофеев, И. Н. Максимова, В. В. Афонин; Национальный исследовательский Мордовский государственный университет им. Н.П. Огарева, Пензенский государственный университет архитектуры и строительства. – Саранск: Национальный исследовательский Мордовский государственный университет им. Н.П. Огарёва, 2019. – 140 с. – ISBN 978-5-7103-3735-6. – EDN ZBQYQX.