

Д. Р. Чуркин^{1✉}, Е. Ю. Воронкин¹

Разработка веб-приложения «Аптечка+» для учета домашних лекарственных средств и контроля срока их годности

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: denis.churkin.afikei@mail.ru

Аннотация. В статье представлены результаты разработки веб-приложения «Аптечка+», предназначенного для учета домашних лекарственных средств и контроля срока их годности. Рассмотрена проблема бессистемного хранения информации о лекарствах в домашних условиях, при котором пользователь может не знать фактическое наличие препаратов, их количество и дату окончания срока годности. Выполнен анализ предметной области и существующих решений для ведения домашней аптечки и напоминаний о приеме лекарственных средств. Для реализации приложения была выбрана клиент-серверная архитектура. Клиентская часть разработана с использованием Vue 3, TypeScript, Vue Router, Pinia и Tailwind CSS. Серверная часть реализована на Node.js с применением Express. Для хранения данных использована база данных SQLite. В результате работы было создано MVP веб-приложения, включающее лендинг, регистрацию, авторизацию, защищенную часть системы и раздел управления лекарственными средствами. Приложение позволяет добавлять, редактировать, удалять и просматривать препараты, а также хранить сведения о дозировке, количестве, схеме приема и сроке годности.

Ключевые слова: веб-приложение, домашняя аптечка, учет лекарств, срок годности, Vue 3, Node.js

D. R. Churkin^{1✉}, E. Yu. Voronkin¹

Development of a web application for home medication accounting

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: denis.churkin.afikei@mail.ru

Abstract. The article presents the results of developing the “Aptechka+” web application designed for accounting home medicines and controlling their expiration dates. The problem of unsystematic storage of information about medicines at home is considered. In such conditions, the user may not know the actual availability of medicines, their quantity, and expiration dates. The subject area and existing solutions for maintaining a home first-aid kit and medication reminders are analyzed. A client-server architecture was chosen for the application implementation. The client side was developed using Vue 3, TypeScript, Vue Router, Pinia, and Tailwind CSS. The server side was implemented using Node.js and Express. SQLite was used for data storage. As a result, an MVP version of the web application was created, including a landing page, registration, authorization, a protected user area, and a medication management section. The application allows users to add, edit, delete, and view medicines, as well as store information about dosage, quantity, intake schedule, and expiration date.

Keywords: web application, home first-aid kit, medication accounting, expiration date, Vue 3, Node.js

Введение

В настоящее время цифровые технологии активно применяются в различных сферах повседневной жизни человека. Одним из направлений их использо-

вания является автоматизация хранения и обработки персональной информации. К такой информации можно отнести сведения о лекарственных средствах, находящихся в домашней аптечке.

Большинство пользователей хранит дома лекарственные препараты для оказания первой помощи, лечения сезонных заболеваний или регулярного применения по назначению врача. При этом учет лекарственных средств часто ведется бессистемно либо не ведется вовсе. Информация о наличии препаратов, их количестве, сроке годности и схеме приема может храниться в бумажном виде, в заметках или запоминаться пользователем самостоятельно.

Такой подход может приводить к различным проблемам. Пользователь может приобрести лекарственное средство повторно, не зная, что оно уже есть дома. Также возможна ситуация, при которой препарат находится в аптечке, но его срок годности уже истек. Кроме того, при большом количестве лекарств затрудняется быстрый поиск нужного препарата и контроль его остатка.

На рынке существуют решения для учета медикаментов и напоминаний о приеме препаратов. Часть таких приложений ориентирована на мобильные устройства, часть предназначена для контроля приема лекарств, а отдельные сервисы содержат избыточный функционал, который не всегда требуется обычному пользователю при ведении домашней аптечки [1–4]. В связи с этим актуальной задачей является разработка простого веб-приложения, позволяющего вести список домашних лекарственных средств и контролировать основные сведения о них.

Целью работы является разработка веб-приложения для учета домашних лекарственных средств и контроля срока их годности. Для достижения цели были поставлены следующие задачи: выполнить анализ предметной области, определить функциональные требования, выбрать технологии разработки, спроектировать структуру базы данных, реализовать клиентскую и серверную части приложения, а также провести тестирование основных пользовательских сценариев.

Исследование предметной области

Домашняя аптечка представляет собой набор лекарственных средств, которые используются пользователем для оказания первой помощи, лечения сезонных заболеваний, выполнения врачебных назначений или хранения препаратов постоянного применения. По мере увеличения количества лекарств возрастает сложность контроля их наличия, количества, срока годности и схемы приема.

Основными данными, которые необходимо учитывать при ведении домашней аптечки, являются наименование лекарственного средства, форма выпуска, дозировка, срок годности, количество препарата и схема приема. Дополнительно может потребоваться хранение признака напоминаний о приеме, чтобы в дальнейшем использовать эти данные для уведомлений пользователя.

При отсутствии систематизированного учета пользователь сталкивается с рядом затруднений. К ним относится невозможность быстро определить наличие нужного препарата, риск хранения и использования просроченных лекарственных средств, а также сложность контроля остатков. Кроме того, при регулярном

приеме медикаментов возникает необходимость напоминаний, позволяющих снизить вероятность пропуска приема.

В ходе анализа существующих решений были рассмотрены приложения, предназначенные для учета домашней аптечки, контроля сроков годности и напоминаний о приеме лекарств. Приложение «Домашняя аптечка» ориентировано на контроль сроков годности лекарств, составление графика приема и получение напоминаний [1]. Приложение «Аптечка. Учет медикаментов» позволяет создавать аптечки, добавлять препараты, искать нужные лекарства и получать уведомления [2]. Отдельную группу составляют решения MyTherapy и Medisafe, которые в большей степени ориентированы на напоминания о приеме лекарств и отслеживание назначений [3, 4].

Проведенный анализ показал, что существующие решения закрывают отдельные задачи учета лекарственных средств. Одни приложения ориентированы на ведение списка аптечки, другие – на напоминания о приеме препаратов, третьи – на предоставление справочной информации. При этом часть решений доступна преимущественно в мобильном формате или содержит функции, которые не всегда требуются обычному пользователю для учета домашней аптечки.

На основании анализа предметной области были определены основные требования к разрабатываемому веб-приложению. Система должна обеспечивать регистрацию и авторизацию пользователей, просмотр списка лекарственных средств, добавление, редактирование и удаление препаратов, поиск и фильтрацию записей, хранение информации о сроке годности и количестве, а также защиту данных пользователя.

Методы и технологии

При разработке веб-приложения была выбрана клиент-серверная архитектура. Такой подход позволяет разделить пользовательский интерфейс, серверную логику и хранение данных. Клиентская часть отвечает за отображение интерфейса и обработку действий пользователя, серверная часть – за обработку запросов, авторизацию и взаимодействие с базой данных.

Для реализации клиентской части был выбран фреймворк Vue 3 [5]. Он позволяет разрабатывать интерфейс на основе компонентного подхода. Такой подход упрощает сопровождение приложения, так как отдельные элементы интерфейса могут быть выделены в самостоятельные компоненты. В качестве языка программирования был использован TypeScript [6], который обеспечивает статическую типизацию и позволяет снизить вероятность ошибок на этапе разработки.

Для сборки клиентского приложения применялся Vite [7]. Маршрутизация между страницами реализована с помощью Vue Router [8]. Для управления состоянием приложения использовалась библиотека Pinia [9]. Она применялась для хранения данных авторизации, информации о пользователе и списка лекарственных средств. Для оформления интерфейса использовался Tailwind CSS [10], а для выполнения HTTP-запросов между клиентской и серверной частями – Axios [11].

Серверная часть приложения была реализована на платформе Node.js [12] с использованием фреймворка Express [13]. Express применялся для создания

REST API, описания маршрутов и обработки HTTP-запросов. Для проверки входных данных использовалась библиотека Zod [14]. Она позволила описать схемы данных для регистрации, авторизации и операций с лекарственными средствами.

Для авторизации пользователя был использован механизм JSON Web Token [15]. После успешного входа сервер формирует токен, который используется при последующих запросах к защищенным маршрутам. Для хранения данных была выбрана SQLite [16, 17]. Данная система управления базами данных подходит для MVP-версии приложения, так как не требует отдельного серверного процесса и позволяет хранить данные в одном файле.

В базе данных были предусмотрены две основные таблицы: `users` и `medications`. Таблица `users` предназначена для хранения учетных записей пользователей. В ней сохраняются идентификатор пользователя, имя, адрес электронной почты, хеш пароля, дата создания и дата обновления записи. Таблица `medications` предназначена для хранения информации о лекарственных средствах. В ней сохраняются идентификатор лекарства, идентификатор пользователя, название препарата, форма выпуска, дозировка, схема приема, количество, начальное количество, срок годности, признак включения напоминаний, дата создания и дата обновления записи.

Между пользователем и лекарственными средствами установлена связь «один ко многим». Один пользователь может иметь несколько записей о лекарствах, но каждая запись относится только к одному пользователю. Это позволяет формировать индивидуальный список препаратов для каждой учетной записи и ограничивать доступ к данным других пользователей.

Проект был организован в формате монорепозитория. В рамках одной структуры проекта были размещены клиентская и серверная части приложения. Такой подход упрощает запуск, сборку и сопровождение проекта.

Результаты

В результате работы было разработано MVP веб-приложения «Аптечка+», предназначенное для ведения домашней аптечки. В приложении реализованы публичная и закрытая части. Публичная часть включает лендинг, страницу регистрации и страницу авторизации. Лендинг предназначен для знакомства пользователя с возможностями сервиса и перехода к регистрации или входу в систему.

Закрытая часть приложения доступна только авторизованному пользователю. При попытке перейти в закрытый раздел без авторизации пользователь перенаправляется на страницу входа. Для этого на клиентской стороне используются навигационные ограничения Vue Router, а на серверной стороне – проверка авторизации при обращении к защищенным маршрутам.

Основным разделом закрытой части является страница «Мои лекарства». В данном разделе пользователь может просматривать список добавленных препаратов, выполнять поиск по наименованию, добавлять новые лекарственные средства, редактировать существующие записи и удалять ненужные препараты.

Пример пользовательского интерфейса раздела «Мои лекарства» представлен на рис. 1.

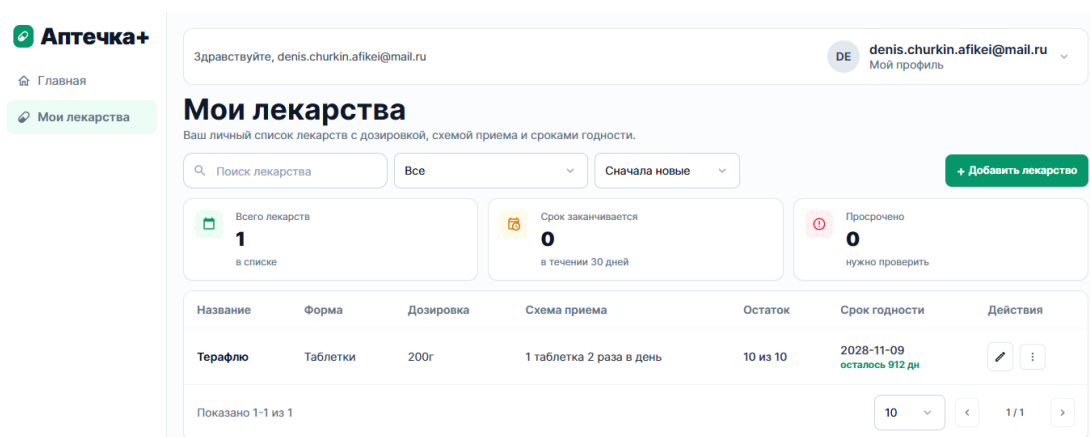


Рис. 1. Интерфейс раздела «Мои лекарства» веб-приложения «Аптечка+»

Для отображения лекарственных средств используется табличное представление. В таблице выводятся основные сведения о препаратах: наименование, форма выпуска, дозировка, схема приема, количество, срок годности и состояние признака напоминаний. Табличное представление позволяет компактно отображать данные и быстро находить необходимую информацию.

Добавление лекарственного средства реализовано через модальное окно с формой. Пользователь указывает название препарата, форму выпуска, дозировку, схему приема, количество, срок годности и признак включения напоминаний. После сохранения данные отправляются на серверную часть приложения и записываются в базу данных.

Редактирование лекарственного средства выполняется с использованием той же формы. При открытии формы поля заполняются текущими данными выбранного препарата. После изменения данных пользователь сохраняет запись, а обновленная информация отображается в списке лекарственных средств. Удаление препарата выполняется из таблицы. После подтверждения действия запись удаляется из базы данных и больше не отображается в интерфейсе.

Серверная часть предоставляет маршруты для регистрации, авторизации, получения текущего пользователя, выхода из учетной записи и управления лекарственными средствами. Основные маршруты API включают /auth/register, /auth/login, /auth/me, /auth/logout, а также маршруты с префиксом /medications, предназначенные для создания, получения, редактирования и удаления записей.

После реализации приложения было проведено тестирование основных пользовательских сценариев. Проверялись регистрация пользователя, авторизация, выход из учетной записи, защита закрытых маршрутов, добавление лекарственного средства, редактирование записи, удаление лекарства, поиск и отображение срока годности. Также проверялась корректность взаимодействия клиентской части с серверным API и сохранение данных в базе данных.

По результатам тестирования критических дефектов выявлено не было. Основные пользовательские сценарии выполнялись корректно, данные сохранялись в базе данных, а доступ к закрытым разделам ограничивался механизмом авторизации.

Заключение

В ходе работы было разработано веб-приложение «Аптечка+» для учета домашних лекарственных средств и контроля срока их годности. В процессе разработки был выполнен анализ предметной области, рассмотрены существующие решения, сформированы функциональные требования, спроектирована база данных и реализованы клиентская и серверная части приложения.

Разработанное приложение позволяет пользователю зарегистрироваться, авторизоваться, просматривать список лекарственных средств, добавлять новые препараты, редактировать и удалять записи, а также хранить сведения о количестве, дозировке, схеме приема и сроке годности. Реализованный функционал соответствует задачам MVP-версии и может быть использован как основа для дальнейшего развития системы.

Практическая значимость работы заключается в том, что приложение может применяться для организации домашней аптечки и упрощения контроля лекарственных средств. Использование такого сервиса позволяет снизить риск хранения просроченных препаратов и повысить удобство поиска информации о лекарствах.

В дальнейшем приложение может быть расширено за счет полноценной системы уведомлений, категорий лекарственных средств, истории приема, адаптивной мобильной версии, синхронизации данных между устройствами и перехода на более производительную серверную базу данных.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Домашняя аптечка : приложение. — URL: <https://www.rustore.ru/catalog/app/ru.application.homemedkit> (дата обращения: 03.05.2026). — Текст: электронный.
2. Аптечка. Учет медикаментов : приложение. — URL: <https://apps.apple.com/by/app/аптечка-учет-медикаментов/id1565373036> (дата обращения: 03.05.2026). — Текст: электронный.
3. MyTherapy : официальный сайт. — URL: <https://www.mytherapyapp.com> (дата обращения: 03.05.2026). — Текст : электронный.
4. Medisafe : официальный сайт. — URL: <https://www.medisafe.com> (дата обращения: 03.05.2026). — Текст : электронный.
5. Vue.js. Introduction : официальная документация. — URL: <https://vuejs.org/guide/introduction> (дата обращения: 03.05.2026). — Текст : электронный.
6. TypeScript Documentation : официальная документация. — URL: <https://www.typescriptlang.org/docs/> (дата обращения: 03.05.2026). — Текст : электронный.
7. Vite. Getting Started : официальная документация. — URL: <https://vite.dev/guide/>
8. (дата обращения: 03.05.2026). — Текст : электронный.
9. Vue Router. The official Router for Vue.js : официальная документация. — URL: <https://router.vuejs.org/> (дата обращения: 03.05.2026). — Текст : электронный.

10. Pinia. The intuitive store for Vue.js : официальная документация. – URL: <https://pinia.vuejs.org/> (дата обращения: 03.05.2026). – Текст : электронный.
11. Tailwind CSS Documentation : официальная документация. – URL: <https://tailwindcss.com/docs> (дата обращения: 03.05.2026). – Текст : электронный.
12. Axios. Getting Started : официальная документация. – URL: <https://axios-http.com/docs/intro> (дата обращения: 03.05.2026). – Текст : электронный.
13. Node.js Documentation : официальная документация. – URL: <https://nodejs.org/docs/latest/api/> (дата обращения: 03.05.2026). – Текст : электронный.
14. Express. Node.js web application framework : официальный сайт. – URL: <https://expressjs.com/> (дата обращения: 03.05.2026). – Текст : электронный.
15. Zod Documentation : официальная документация. – URL: <https://zod.dev/> (дата обращения: 03.05.2026). – Текст : электронный.
16. JSON Web Tokens. Introduction to JSON Web Tokens : официальный сайт. – URL: <https://jwt.io/introduction> (дата обращения: 03.05.2026). – Текст : электронный.
17. SQLite Documentation : официальная документация. – URL: <https://sqlite.org/docs.html> (дата обращения: 03.05.2026). – Текст : электронный.
18. SQLite. About SQLite : официальный сайт. – URL: <https://sqlite.org/about.html> (дата обращения: 03.05.2026). – Текст : электронный.
19. Флэнаган, Д. JavaScript. Подробное руководство. 7-е изд. – Санкт-Петербург : Питер, 2021. – 720 с.
20. Бэнкс, А. Изучаем JavaScript. Современные возможности языка / А. Бэнкс, Е. Порселло. – Санкт-Петербург : Питер, 2021. – 320 с.
21. Хэнчетт, Э. Vue.js в действии / Э. Хэнчетт, Б. Листуон. – Москва : ДМК Пресс, 2019. – 304 с.

© Д. Р. Чуркин, Е. Ю. Воронкин, 2026