

А. С. Панов¹✉, С. Ю. Кацко¹

Разработка веб-приложения для управления личными задачами с системой анализа и рекомендаций по повышению продуктивности

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: vip.zip.rar@mail.ru

Аннотация. В статье представлены результаты создания веб-приложения для персонального управления задачами, включающего модуль статистики и генерации рекомендаций. Проведен анализ существующих методик (GTD, Pomodoro, матрица Эйзенхауэра) и программных аналогов (Todoist, TickTick, Microsoft To Do), выявлен дефицит полностью бесплатных локальных решений с аналитикой. Предложена модульная архитектура на чистом JavaScript с хранением данных в LocalStorage. Реализованы CRUD-операции, фильтрация, сортировка, визуализация динамики (Chart.js) и продукционная модель рекомендаций (4 правила: низкий процент выполнения, просроченные высокоприоритетные задачи, вечерняя активность, отставание по категориям). Проведено тестирование (Jest, ручное, Lighthouse): производительность 98 баллов, лучшие практики 100. Приложение не требует регистрации, работает полностью на стороне клиента.

Ключевые слова: веб-приложение, управление задачами, продуктивность, LocalStorage, JavaScript, продукционная модель, визуализация данных, тестирование

A. S. Panov¹✉, S. Yu. Katsko¹

Development of a web application for managing personal tasks with a system of analysis and recommendations for increasing productivity

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: vip.zip.rar@mail.ru

Abstract. The article presents the results of developing a web application for personal task management that includes a statistics module and a recommendation generator. An analysis of existing methodologies (GTD, Pomodoro, Eisenhower matrix) and software analogues (Todoist, TickTick, Microsoft To Do) was carried out, revealing a shortage of completely free local solutions with analytics. A modular architecture using pure JavaScript with LocalStorage data storage is proposed. CRUD operations, filtering, sorting, dynamics visualisation (Chart.js) and a production rule-based recommendation model (4 rules: low completion rate, overdue high-priority tasks, evening activity, lagging categories) are implemented. Testing (Jest, manual, Lighthouse) shows performance of 98 points and best practices of 100. The application requires no registration and runs entirely on the client side.

Keywords: web application, task management, productivity, LocalStorage, JavaScript, production model, data visualisation, testing

Введение

В условиях информационной перегрузки и многозадачности управление личными делами становится критически важным. Бумажные списки уступают

место цифровым инструментам, однако многие из них либо требуют платной подписки для доступа к аналитике, либо хранят данные на серверах, что вызывает опасения о конфиденциальности. Существует потребность в бесплатном, полностью локальном инструменте, сочетающем планирование задач с обратной связью по продуктивности и персонализированными советами.

Цель работы – разработка веб-приложения для управления личными задачами с модулем анализа и системой рекомендаций. Задачи: анализ предметной области и аналогов, проектирование архитектуры и пользовательского интерфейса, реализация на JavaScript с LocalStorage, алгоритмизация рекомендаций на производственных правилах, тестирование.

Исследование предметной области

Исследования показывают, что методики GTD (сбор, обработка, организация, обзор, выполнение) [1], Pomodoro (25-минутные интервалы) [2] и матрица Эйзенхауэра [3] могут быть эффективно поддержаны программными средствами. Из GTD вытекают требования к единому хранилищу задач, категоризации и инструментам обзора (фильтрация, сортировка). Метод Pomodoro обосновывает включение рекомендации по интервальной работе при низком проценте выполнения. Матрица Эйзенхауэра в упрощенной трехуровневой проекции (низкий, средний, высокий приоритет) служит основой для визуальной индикации и анализа просроченных задач.

Проведенный обзор существующих решений Todoist, TickTick, Microsoft To Do, Trello, Яндекс Трекер [4–8] выявил, что аналитические функции либо платные, либо отсутствуют, а хранение данных везде серверное с обязательной регистрацией. Ни один из продуктов не предоставляет персонализированных рекомендаций на основе поведения пользователя. Это обосновывает актуальность разработки полностью локального бесплатного приложения с аналитикой и рекомендательным модулем.

Методы и технологии

При разработке применялась клиентская архитектура (Single Page Application) без серверной части. Основной язык – JavaScript (ES6+) [9], среда разработки – Visual Studio Code [10], тестирование – Jest [11] и Chrome DevTools [12], визуализация данных – библиотека Chart.js.

Хранение данных реализовано через LocalStorage. Сериализация в JSON, ключ «tasks». Структура задачи: id, title, description, category, priority (low/medium/high), dueDate, isCompleted, createdAt, completedAt. Обрабатывается ситуация переполнения квоты (5 МБ) и недоступности хранилища.

Модульная архитектура включает пять компонентов:

- TaskManager – CRUD, фильтрация, сортировка.
- StorageService – чтение/запись LocalStorage.
- RenderService – динамическое обновление DOM.
- AnalyticsService – расчет процента выполнения, просроченных, распределения по категориям, динамики по дням.

– RecommendationEngine – генерация советов на основе правил.

Правила (пороги обоснованы литературой по тайм-менеджменту):

1. Если процент выполнения за последние 7 дней $< 40\%$ → предложить метод Pomodoro.

2. Если невыполненных задач с высоким приоритетом и просроченным сроком > 2 → пересмотреть сроки или делегировать.

3. Если доля завершённых после 18:00 или до 6:00 $> 60\%$ → планировать важное на утро.

4. Если по какой-либо категории процент выполнения $< 30\%$ (при наличии не менее 3 задач) → уделить внимание этой категории.

Функция generateRecommendations() вызывается после каждого изменения данных, возвращает не более трех советов.

Результаты

Разработано веб-приложение, доступное в браузере без установки. Интерфейс (рис. 1) имеет два режима: «Задачи» и «Статистика». В режиме задач реализованы:

- создание / редактирование / удаление задач через модальное окно;
- визуальная индикация приоритетов (зеленый – низкий, желтый – средний, красный – высокий);
- фильтрация по категории (Работа, Учеба, Дом, Личное) и приоритету;
- сортировка по дате создания, сроку или приоритету;
- отметка выполнения с фиксацией времени.

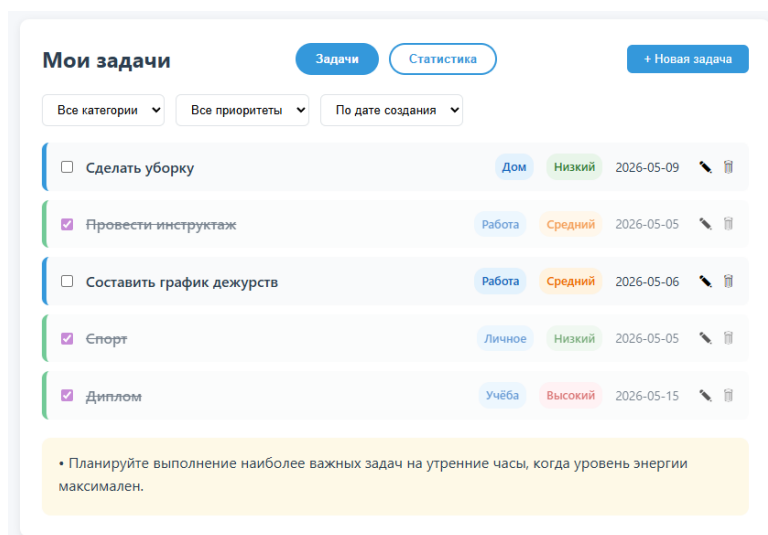


Рис. 1. Главный экран приложения

Режим статистики (рис. 2) отображает:

- общее количество задач, процент выполнения, число просроченных;
- линейный график динамики созданных и выполненных задач по дням (за неделю, месяц или все время);

– круговую диаграмму распределения задач по категориям.

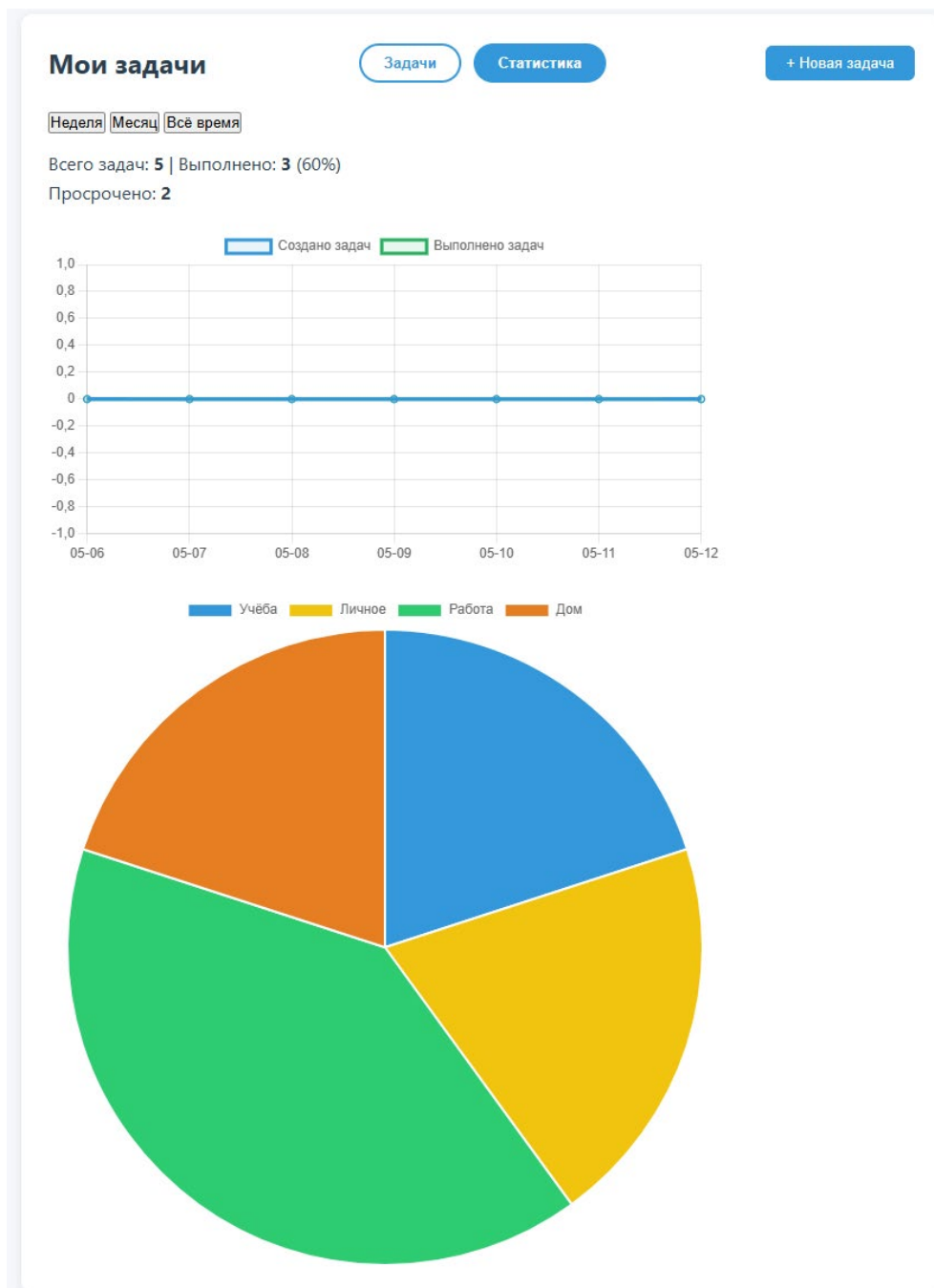


Рис. 2. Страница статистики

Блок рекомендаций динамически обновляется под списком задач. Примеры срабатывания правил: при падении недельной продуктивности ниже 40 % выводится совет использовать Pomodoro; при накоплении трех просроченных высокоприоритетных задач – рекомендация пересмотреть планы.

Модульное тестирование (Jest [11]) подтвердило корректность функций фильтрации, сортировки, расчета статистики и логики рекомендаций. Ручное функциональное тестирование охватило все пользовательские сценарии: создание, ре-

дактирование, удаление, отметку выполнения, сохранность после перезагрузки, адаптивность (320–1920 пикселей) с использованием браузерных инструментов разработчика [12, 13]. Аудит Lighthouse [14] дал следующие результаты: производительность – 98, доступность – 75, лучшие практики – 100, SEO – 90 (рис. 3). Недостатки доступности связаны с отсутствием текстовых меток у символьных кнопок и могут быть устранены в следующих версиях.

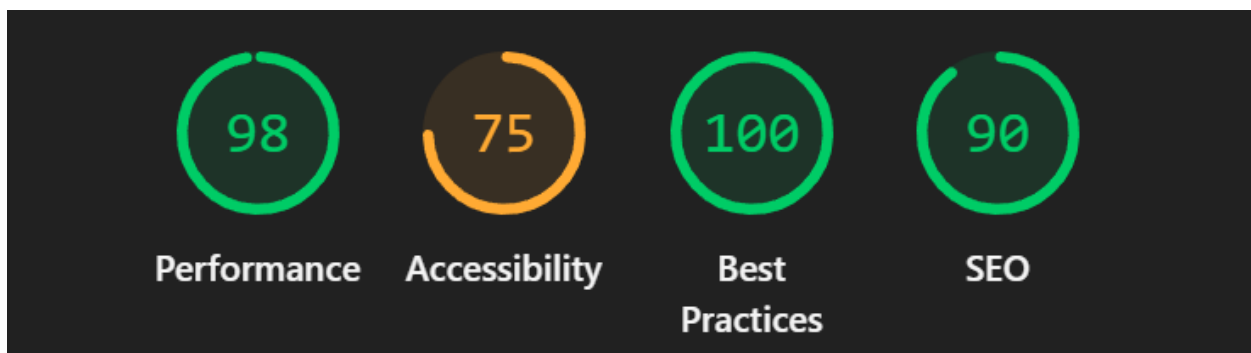


Рис. 3. Результаты Lighthouse

Заключение

В ходе работы разработано клиентское веб-приложение для управления личными задачами с системой анализа и рекомендаций. Основные достижения:

- проведен анализ предметной области и методик продуктивности;
- предложена модульная архитектура на JavaScript с хранилищем LocalStorage;
- реализованы CRUD, фильтрация, сортировка, визуализация статистики (Chart.js);
- создана производственная модель генерации рекомендаций (4 правила);
- выполнено тестирование (Jest, ручное, Lighthouse), подтвердившее высокую производительность и соответствие требованиям.

Приложение может использоваться студентами и офисными сотрудниками для повышения личной эффективности. Перспективы развития: добавление синхронизации между устройствами (серверный слой), расширение набора производственных правил, улучшение доступности интерфейса.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Аллен Д. Как привести дела в порядок: искусство продуктивности без стресса / Д. Аллен ; пер. с англ. Ю. Константиновой. – М. : Манн, Иванов и Фербер, 2015. – 416 с.
2. Чирилло Ф. Метод Pomodoro: управление временем / Ф. Чирилло ; пер. с англ. – М. : Эксмо, 2022. – 224 с.
3. Кови С. 7 навыков высокоэффективных людей / С. Кови ; пер. с англ. – 18 е изд. – М. : Альпина Паблишер, 2023. – 396 с.
4. Todoist: приложение для управления задачами. – URL: <https://todoist.com/ru> (дата обращения: 16.04.2026). – Текст : электронный.
5. TickTick: To Do List & Calendar. – URL: <https://ticktick.com/> (дата обращения: 16.04.2026). – Текст : электронный.

6. Microsoft To Do. – URL: <https://to-do.office.com/> (дата обращения: 16.04.2026). – Текст : электронный.
7. Trello: Управление заданиями. – URL: <https://trello.com/ru/use-cases/task-management> (дата обращения: 16.04.2026). – Текст : электронный.
8. Яндекс Трекер. – URL: <https://tracker.yandex.ru/> (дата обращения: 16.04.2026). – Текст : электронный.
9. Флэнаган Д. JavaScript. Полное руководство / Д. Флэнаган ; пер. с англ. А. А. Слинкина. – 7 е изд. – СПб. : Диалектика, 2021. – 720 с.
10. Visual Studio Code. – URL: <https://code.visualstudio.com/> (дата обращения: 21.04.2026). – Текст : электронный.
11. Jest: Delightful JavaScript Testing Framework. – URL: <https://jestjs.io/> (дата обращения: 21.04.2026). – Текст : электронный.
12. Chrome DevTools: отладка JavaScript // Chrome for Developers. – URL: <https://developer.chrome.google.cn/docs/devtools/javascript> (дата обращения: 21.04.2026). – Текст : электронный.
13. Общие сведения о средствах разработки // Microsoft Edge Developer documentation. – URL: <https://learn.microsoft.com/ru-ru/microsoft-edge/devtools/overview> (дата обращения: 21.04.2026). – Текст : электронный.
14. Lighthouse: Оптимизируйте свой сайт // Chrome for Developers. – URL: <https://developer.chrome.google.cn/docs/devtools/lighthouse> (дата обращения: 21.04.2026). – Текст : электронный.

© А. С. Панов, С. Ю. Кацко, 2026