

КАК СОЗДАТЬ

ПЛАНИРОВЩИК СЕМЕЙНОГО БЮДЖЕТА



ДИАНА ВАСИЛЬЕВА

Диана Васильева

**Как создать планировщик
семейного бюджета**

«Автор»

2026

Васильева Д.

Как создать планировщик семейного бюджета / Д. Васильева —
«Автор», 2026

Хотите видеть, куда уходят деньги, но сложные таблицы быстро забрасываются? В этой книге вы создадите собственный планировщик семейного бюджета: быстрый ввод доходов и расходов, месячную сводку, лимиты категорий, предупреждения, накопительные цели и безопасный резерв. Все шаги рассчитаны на новичка: готовые запросы для ИИ-помощника, понятные примеры, проверка формул и защита от ошибочного ввода. Вы протестируете пересчёт после исправлений, операции разных месяцев, мобильный экран и сохранение данных. Подойдёт семьям и самозанятым. В конце — версии для личного, семейного и проектного учёта, способы адаптации и приглашение в клуб практического вайбкодинга.

© Васильева Д., 2026

© Автор, 2026

Содержание

Как работать с этой книгой	6
Что именно мы создаём	7
Подготовка аккаунта и проекта	8
Техническое задание на человеческом языке	9
Запрос 1. Создаём каркас	10
Запрос 2. Добавляем: быстрое добавление дохода или расхода	11
Запрос 3. Добавляем: месячная сводка	12
Запрос 4. Добавляем: лимиты по категориям	13
Запрос 5. Добавляем: предупреждение о приближении к лимиту	14
Запрос 6. Добавляем: накопительные цели	15
Запрос 7. Добавляем: редактирование ошибочной операции	16
Запрос 8. Добавляем: экспорт понятной текстовой сводки	17
Сценарий пользователя от начала до результата	18
Конец ознакомительного фрагмента.	19

Диана Васильева

Как создать планировщик семейного бюджета

Как создать планировщик семейного бюджета

Создаём в [Chatium](#) учёт доходов, расходов, лимитов, целей и безопасного остатка

Диана Васильева

Практическое руководство для новичков · 2026

Результат книги: вы создадите веб-приложение для внесения операций, категорий, месячных лимитов и накопительных целей.

Как работать с этой книгой

Эта книга устроена как мастерская. Не нужно сначала читать её целиком. Откройте [Chatium](#), держите книгу рядом и выполняйте шаги по порядку.

В каждом шаге есть пять элементов: действие, готовый запрос, ожидаемый результат, проверка и команда для исправления. Если интерфейс платформы обновился и название кнопки изменилось, ориентируйтесь на смысл действия.

Не пытайтесь получить идеальное приложение одним длинным запросом. Вайбкодинг работает лучше, когда вы создаёте небольшой рабочий слой, проверяете его и только потом добавляете следующий.

Сохраните отдельную копию проекта перед крупными изменениями. Никогда не вставляйте в запросы пароли, данные банковских карт, документы клиентов и другие чувствительные сведения.

Что именно мы создаём

Итоговый продукт — веб-приложение для внесения операций, категорий, месячных лимитов и накопительных целей. Он будет особенно полезен семьям, самозанятым и всем, кто хочет видеть деньги без сложных таблиц.

У решения будет простой первый экран, понятный сценарий пользователя, проверка введённых данных и сообщение о результате. Мы не будем добавлять функции только ради впечатления.

Ключевой принцип: $\text{Остаток} = \text{доходы} - \text{расходы}$. Безопасный остаток = остаток – обязательный резерв. Прогресс цели = $\text{накоплено} / \text{цель} \times 100\%$.

Сначала соберём минимальную рабочую версию. Затем улучшим оформление, сохранение данных, мобильный вид, безопасность и способы использования.

Подготовка аккаунта и проекта

Перейдите в [Chatium](#) по ссылке из книги. Создайте аккаунт или войдите в существующий. Один аккаунт платформы можно воспринимать как отдельное пространство: у него собственный адрес, код, таблицы и пользователи.

Создайте отдельный проект для этой книги. Назовите его коротко и понятно, латинскими буквами, без пробелов. Не используйте рабочий проект клиента для экспериментов.

Откройте встроенную среду разработки и помощника. В зависимости от версии интерфейса это может быть веб-редактор, раздел разработки или окно ИИ-помощника.

Перед первым запросом опишите, что вы новичок и хотите получать изменения небольшими шагами. Это снизит риск, что помощник сразу создаст слишком сложную архитектуру.

Техническое задание на человеческом языке

Хорошее техническое задание не обязано быть длинным. Оно должно отвечать на вопросы: кто пользуется инструментом, что вводит, что получает, какие ошибки нельзя допустить и где хранятся данные.

Поля первой версии: Дата операции — дата; Тип — доход или расход; Категория — список; Сумма — положительное число; Комментарий — короткий текст; Лимит категории — сумма на месяц; Цель — название, сумма и срок.

Мы отделяем обязательные поля от необязательных. Для каждого числа задаём допустимый диапазон, для списка — варианты, для даты — ограничения.

Скопируйте запрос ниже в помощник [Chatium](#). Не сокращайте его до одной строки: контекст помогает получить более управляемый результат.

Помоги уточнить техническое задание для инструмента «планировщик семейного бюджета». Кому он будет полезен: семьям, самозанятым и всем, кто хочет видеть деньги без сложных таблиц. Результат: веб-приложение для внесения операций, категорий, месячных лимитов и накопительных целей. Поля: Дата операции — дата; Тип — доход или расход; Категория — список; Сумма — положительное число; Комментарий — короткий текст; Лимит категории — сумма на месяц; Цель — название, сумма и срок. Формула или правило: Остаток = доходы – расходы. Безопасный остаток = остаток – обязательный резерв. Прогресс цели = накоплено / цель × 100%. Задай мне не больше семи вопросов. Не предлагай код, пока я не отвечу.

Запрос 1. Создаём каркас

На этом шаге нужен не красивый продукт, а работающий маршрут и экран без ошибок. Просим помощника объяснить структуру файлов и не подключать лишние зависимости.

После выполнения откройте адрес проекта в новой вкладке. Если экран загружается, название видно и кнопка реагирует — каркас готов.

Если вместо приложения появляется ошибка, не просите «исправить всё». Передайте помощнику точный текст ошибки и попросите изменить только необходимый файл.

Я новичок. Создай в [Chatium](#) минимальную рабочую версию инструмента «планировщик семейного бюджета». Пользователь должен получить следующий результат: веб-приложение для внесения операций, категорий, месячных лимитов и накопительных целей. Сначала перечисли файлы и кратко объясни архитектуру. Затем создай главный маршрут и один экран. Не подключай внешние сервисы и не добавляй функции, которых я не просила. Критерий готовности: страница открывается по ссылке, виден заголовок и основная кнопка.

Проверка: откройте ссылку в отдельной вкладке и убедитесь, что главный экран загружается без технического текста.

Запрос 2. Добавляем: быстрое добавление дохода или расхода

Теперь добавим функцию «быстрое добавление дохода или расхода». Она должна решать одну понятную задачу и не ломать уже работающие части.

Сначала попросите помощника перечислить файлы, которые он собирается изменить, и кратко объяснить логику. Затем разрешите выполнить изменение.

После обновления проверьте счастливый сценарий, пустые значения и одно заведомо неправильное значение. Не переходите дальше, пока результат не объясним.

Если помощник меняет дизайн, названия полей или формулу без запроса, верните предыдущую рабочую версию и повторите задачу с жёстким ограничением: «не менять существующую логику».

В текущем проекте [Chatium](#) добавь функцию «быстрое добавление дохода или расхода». Не меняй существующие названия полей, формулу и рабочие маршруты. До изменения перечисли затрагиваемые файлы. После изменения опиши три ручных теста. Если требуется новое поле данных, сначала объясни его название, тип и обязательность.

Критерий готовности: функция «быстрое добавление дохода или расхода» работает в основном и ошибочном сценарии, а предыдущие функции не изменились.

Запрос 3. Добавляем: месячная сводка

Теперь добавим функцию «месячная сводка». Она должна решать одну понятную задачу и не ломать уже работающие части.

Сначала попросите помощника перечислить файлы, которые он собирается изменить, и кратко объяснить логику. Затем разрешите выполнить изменение.

После обновления проверьте счастливый сценарий, пустые значения и одно заведомо неправильное значение. Не переходите дальше, пока результат не объясним.

Если помощник меняет дизайн, названия полей или формулу без запроса, верните предыдущую рабочую версию и повторите задачу с жёстким ограничением: «не менять существующую логику».

В текущем проекте [Chatium](#) добавь функцию «месячная сводка». Не меняй существующие названия полей, формулу и рабочие маршруты. До изменения перечисли затрагиваемые файлы. После изменения опиши три ручных теста. Если требуется новое поле данных, сначала объясни его название, тип и обязательность.

Критерий готовности: функция «месячная сводка» работает в основном и ошибочном сценарии, а предыдущие функции не изменились.

Запрос 4. Добавляем: лимиты по категориям

Теперь добавим функцию «лимиты по категориям». Она должна решать одну понятную задачу и не ломать уже работающие части.

Сначала попросите помощника перечислить файлы, которые он собирается изменить, и кратко объяснить логику. Затем разрешите выполнить изменение.

После обновления проверьте счастливый сценарий, пустые значения и одно заведомо неправильное значение. Не переходите дальше, пока результат не объясним.

Если помощник меняет дизайн, названия полей или формулу без запроса, верните предыдущую рабочую версию и повторите задачу с жёстким ограничением: «не менять существующую логику».

В текущем проекте [Chatium](#) добавь функцию «лимиты по категориям». Не меняй существующие названия полей, формулу и рабочие маршруты. До изменения перечисли затрагиваемые файлы. После изменения опиши три ручных теста. Если требуется новое поле данных, сначала объясни его название, тип и обязательность.

Критерий готовности: функция «лимиты по категориям» работает в основном и ошибочном сценарии, а предыдущие функции не изменились.

Запрос 5. Добавляем: предупреждение о приближении к лимиту

Теперь добавим функцию «предупреждение о приближении к лимиту». Она должна решать одну понятную задачу и не ломать уже работающие части.

Сначала попросите помощника перечислить файлы, которые он собирается изменить, и кратко объяснить логику. Затем разрешите выполнить изменение.

После обновления проверьте счастливый сценарий, пустые значения и одно заведомо неправильное значение. Не переходите дальше, пока результат не объясним.

Если помощник меняет дизайн, названия полей или формулу без запроса, верните предыдущую рабочую версию и повторите задачу с жёстким ограничением: «не менять существующую логику».

В текущем проекте [Chatium](#) добавь функцию «предупреждение о приближении к лимиту». Не меняй существующие названия полей, формулу и рабочие маршруты. До изменения перечисли затрагиваемые файлы. После изменения опиши три ручных теста. Если требуется новое поле данных, сначала объясни его название, тип и обязательность.

Критерий готовности: функция «предупреждение о приближении к лимиту» работает в основном и ошибочном сценарии, а предыдущие функции не изменились.

Запрос 6. Добавляем: накопительные цели

Теперь добавим функцию «накопительные цели». Она должна решать одну понятную задачу и не ломать уже работающие части.

Сначала попросите помощника перечислить файлы, которые он собирается изменить, и кратко объяснить логику. Затем разрешите выполнить изменение.

После обновления проверьте счастливый сценарий, пустые значения и одно заведомо неправильное значение. Не переходите дальше, пока результат не объясним.

Если помощник меняет дизайн, названия полей или формулу без запроса, верните предыдущую рабочую версию и повторите задачу с жёстким ограничением: «не менять существующую логику».

В текущем проекте [Chatium](#) добавь функцию «накопительные цели». Не меняй существующие названия полей, формулу и рабочие маршруты. До изменения перечисли затрагиваемые файлы. После изменения опиши три ручных теста. Если требуется новое поле данных, сначала объясни его название, тип и обязательность.

Критерий готовности: функция «накопительные цели» работает в основном и ошибочном сценарии, а предыдущие функции не изменились.

Запрос 7. Добавляем: редактирование ошибочной операции

Теперь добавим функцию «редактирование ошибочной операции». Она должна решать одну понятную задачу и не ломать уже работающие части.

Сначала попросите помощника перечислить файлы, которые он собирается изменить, и кратко объяснить логику. Затем разрешите выполнить изменение.

После обновления проверьте счастливый сценарий, пустые значения и одно заведомо неправильное значение. Не переходите дальше, пока результат не объясним.

Если помощник меняет дизайн, названия полей или формулу без запроса, верните предыдущую рабочую версию и повторите задачу с жёстким ограничением: «не менять существующую логику».

В текущем проекте [Chatium](#) добавь функцию «редактирование ошибочной операции». Не меняй существующие названия полей, формулу и рабочие маршруты. До изменения перечисли затрагиваемые файлы. После изменения опиши три ручных теста. Если требуется новое поле данных, сначала объясни его название, тип и обязательность.

Критерий готовности: функция «редактирование ошибочной операции» работает в основном и ошибочном сценарии, а предыдущие функции не изменились.

Запрос 8. Добавляем: экспорт понятной текстовой сводки

Теперь добавим функцию «экспорт понятной текстовой сводки». Она должна решать одну понятную задачу и не ломать уже работающие части.

Сначала попросите помощника перечислить файлы, которые он собирается изменить, и кратко объяснить логику. Затем разрешите выполнить изменение.

После обновления проверьте счастливый сценарий, пустые значения и одно заведомо неправильное значение. Не переходите дальше, пока результат не объясним.

Если помощник меняет дизайн, названия полей или формулу без запроса, верните предыдущую рабочую версию и повторите задачу с жёстким ограничением: «не менять существующую логику».

В текущем проекте [Chatium](#) добавь функцию «экспорт понятной текстовой сводки». Не меняй существующие названия полей, формулу и рабочие маршруты. До изменения перечисли затрагиваемые файлы. После изменения опиши три ручных теста. Если требуется новое поле данных, сначала объясни его название, тип и обязательность.

Критерий готовности: функция «экспорт понятной текстовой сводки» работает в основном и ошибочном сценарии, а предыдущие функции не изменились.

Сценарий пользователя от начала до результата

До доработки интерфейса пройдите весь путь так, будто впервые видите приложение. Сценарий нужен не разработчику, а обычному человеку: в нём нет названий файлов и технических команд.

Пользователь добавляет доход или расход сразу после операции.

Каждая операция получает категорию, дату и короткий комментарий.

Главный экран показывает доходы, расходы, остаток и безопасный резерв.

Лимиты предупреждают о приближении к границе, но не запрещают реальную покупку.

Накопительная цель показывает прогресс отдельно от денег на повседневные расходы.

Теперь закройте проект и попробуйте пересказать путь одним предложением. Если для результата приходится объяснять дополнительные действия устно, интерфейс ещё недостаточно понятен.

Отдельно запишите момент, когда пользователь может передумать, ошибиться или вернуться назад. Для такого места нужна безопасная отмена, понятная подсказка или подтверждение.

Конец ознакомительного фрагмента.

Текст предоставлен ООО «Литрес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на Литрес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.