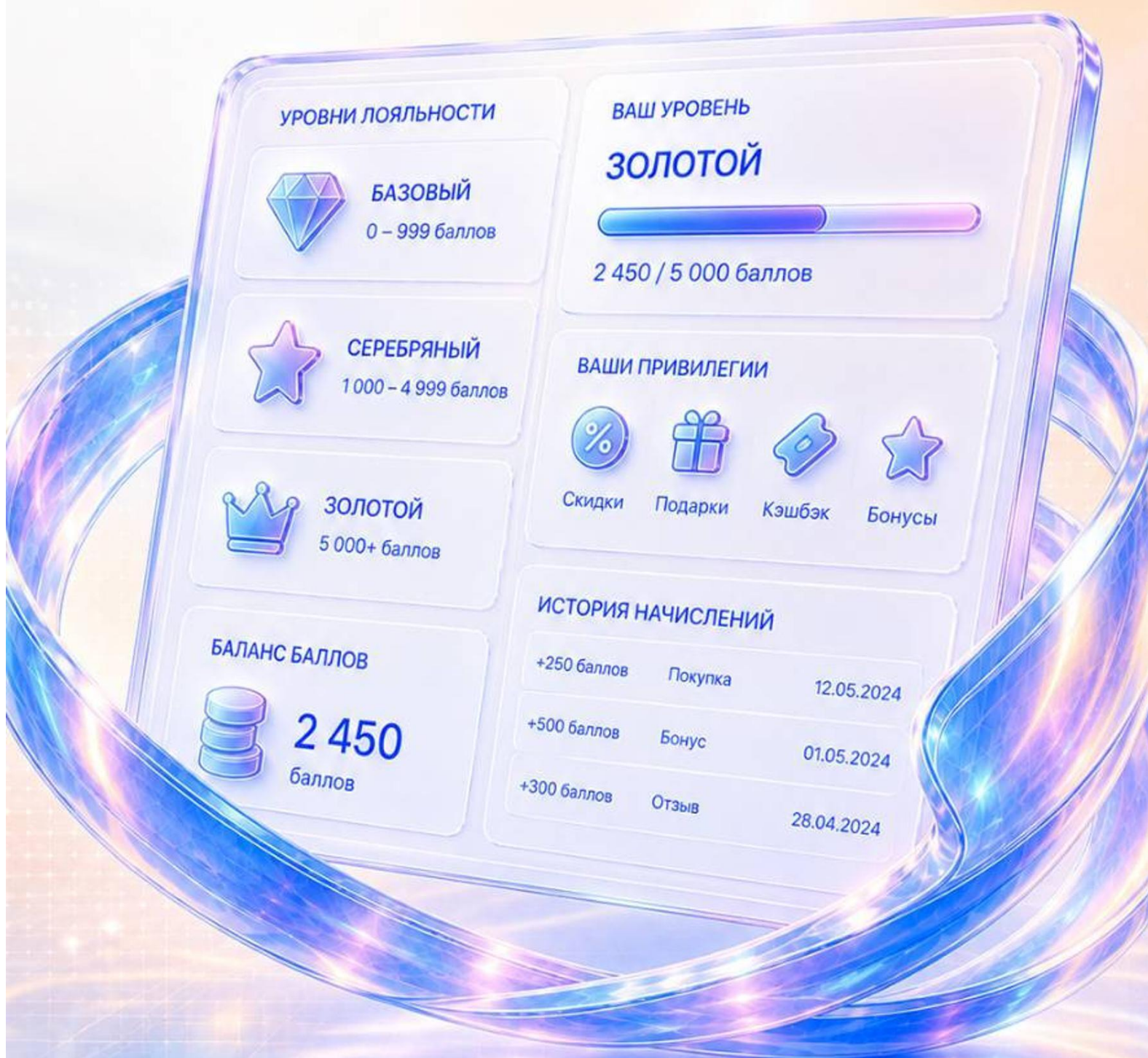


КАК СОЗДАТЬ

ПРОГРАММУ ЛОЯЛЬНОСТИ



ДИАНА ВАСИЛЬЕВА

Диана Васильева

**Как создать
программу лояльности**

«Автор»

2026

Васильева Д.

Как создать программу лояльности / Д. Васильева — «Автор», 2026

Хотите, чтобы клиенты возвращались снова и снова, рекомендовали вас друзьям и покупали чаще? Тогда одной хорошей услуги уже недостаточно – нужна продуманная программа лояльности. Эта книга поможет шаг за шагом создать систему, которая действительно работает. Вы узнаете, как выбрать подходящую механику, начислять бонусы, мотивировать клиентов совершать повторные покупки, внедрить систему без сложных технических решений и избежать распространённых ошибок. Внутри – понятные схемы, готовые примеры, практические рекомендации и инструменты, которые можно адаптировать практически для любого бизнеса: от экспертов и онлайн-школ до салонов красоты, клиник, агентств и интернет-магазинов. Создайте программу лояльности, которая увеличивает прибыль, укрепляет доверие клиентов и помогает вашему бизнесу расти.

© 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: регистрация участника без лишних данных	11
Запрос 3. Добавляем: начисление баллов по прозрачным правилам	12
Запрос 4. Добавляем: каталог наград и проверка доступности	13
Конец ознакомительного фрагмента.	14

Диана Васильева

Как создать программу лояльности

Как создать программу лояльности

Создаём в [Chatium](#) понятную систему баллов, уровней, наград и повторных обращений

Диана Васильева

Практическое руководство для новичков · 2026

Результат книги: вы создадите веб-приложение, где клиент видит баланс баллов и доступные награды, а владелец управляет правилами начисления, списания и уровнями участия.

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, где клиент видит баланс баллов и доступные награды, а владелец управляет правилами начисления, списания и уровнями участия. Он будет особенно полезен мастерам, экспертам, небольшим магазинам, студиям, клубам и локальному бизнесу.

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: Новый баланс = предыдущий баланс + подтверждённые начисления – подтверждённые списания; каждая операция сохраняется в истории и не может изменить баланс дважды.

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: Участник — имя и безопасный идентификатор; Баланс баллов — целое неотрицательное число; Операция — начисление, списание или корректировка; Основание — покупка, визит, рекомендация или ручная причина; Награда — название, стоимость в баллах и доступность; Уровень — стартовый, постоянный или особый; Срок действия — необязательная дата с предупреждением.

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 «программа лояльности». Кому он будет полезен: мастерам, экспертам, небольшим магазинам, студиям, клубам и локальному бизнесу. Результат: веб-приложение, где клиент видит баланс баллов и доступные награды, а владелец управляет правилами начисления, списания и уровнями участия. Поля: Участник — имя и безопасный идентификатор; Баланс баллов — целое неотрицательное число; Операция — начисление, списание или корректировка; Основание — покупка, визит, рекомендация или ручная причина; Награда — название, стоимость в баллах и доступность; Уровень — стартовый, постоянный или особый; Срок действия — необязательная дата с предупреждением. Формула или правило: Новый баланс = предыдущий баланс + подтверждённые начисления – подтверждённые списания; каждая операция сохраняется в истории и не может изменить баланс дважды. 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 «программа лояльности». Пользователь должен получить следующий результат: веб-приложение, где клиент видит баланс баллов и доступные награды, а владелец управляет правилами начисления, списания и уровнями участия. 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: регистрация участника без лишних данных

Теперь добавим функцию «регистрация участника без лишних данных». 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 «регистрация участника без лишних данных». 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 «регистрация участника без лишних данных» работает в основном и ошибочном сценарии, а предыдущие функции не изменились.

Запрос 3. Добавляем: начисление баллов по прозрачным правилам

Теперь добавим функцию «начисление баллов по прозрачным правилам». 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 «начисление баллов по прозрачным правилам». 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 «начисление баллов по прозрачным правилам» работает в основном и ошибочном сценарии, а предыдущие функции не изменились.

Запрос 4. Добавляем: каталог наград и проверка доступности

Теперь добавим функцию «каталог наград и проверка доступности». 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 «каталог наград и проверка доступности». 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 «каталог наград и проверка доступности» работает в основном и ошибочном сценарии, а предыдущие функции не изменились.

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.