

**Генерация адаптивных веб-приложений для полиграфических задач
с использованием больших языковых моделей
в архитектуре Zero-Dependency**

Пайдершев Алексей Константинович

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В статье рассматривается практическое применение больших языковых моделей для создания специализированного веб-приложения по генерации кувертных карт для мероприятий. Исследование выполнено в рамках парадигмы Zero-Dependency – полного отказа от внешних библиотек и фреймворков. С помощью методологии итеративного промпт-инжиниринга реализовано многофункциональное приложение, позволяющее создавать карты трёх различных форматов (90×110 мм, 195×110 мм, 195×165 мм) с автоматической раскладкой на листы формата А4. Особое внимание уделено решению задач точной типографики, работе с физическими размерами в миллиметрах и генерации двусторонних развёрток для последующей печати и сгиба. Результаты демонстрируют способность искусственного интеллекта эффективно справляться с задачами, требующими прецизионной работы с размерами, автоматизации рутинных операций и создания интуитивного пользовательского интерфейса. Сделан вывод о высокой перспективности применения генеративных моделей для разработки узкоспециализированных полиграфических инструментов.

Ключевые слова: Qwen Studio, большие языковые модели, промпт-инжиниринг, веб-разработка, Zero-Dependency.

**Generating responsive web applications for printing tasks using large
language models in a Zero-Dependency architecture**

Paydershev Alexey Konstantinovich

Sholom-Aleichem Priamursky State University

Student

Abstract

This article examines the practical application of large language models to create a specialized web application for generating event place cards. The study was conducted within the Zero-Dependency paradigm—a complete rejection of external libraries and frameworks. Using an iterative prompt engineering methodology, a multifunctional application was developed that allows for the creation of place cards in three different formats (90x110 mm, 195x110 mm, and 195x165 mm) with automatic layout on A4 sheets. Particular attention is paid to solving problems of

precise typography, working with physical dimensions in millimeters, and generating double-sided layouts for subsequent printing and folding. The results demonstrate the ability of artificial intelligence to effectively handle tasks requiring precise dimensional work, automation of routine operations, and the creation of an intuitive user interface. It is concluded that generative models hold great promise for developing highly specialized printing tools.

Keywords: Qwen Studio, large language models, prompt engineering, web development, Zero-Dependency.

1 Введение

1.1 Актуальность

Организация мероприятий различного масштаб, от камерных ужинов до крупных конференций, неизменно требует решения полиграфических задач, среди которых особое место занимают кувертные карты. Эти небольшие карточки с именами гостей выполняют двойную функцию: помогают участникам найти своё место и добавляют мероприятию элемент торжественности. Традиционно создание таких карт либо заказывается в типографиях, что увеличивает бюджет и время подготовки, либо выполняется вручную в графических редакторах, что требует специальных навыков и значительных временных затрат.

Современные веб-технологии предоставляют возможность автоматизировать этот процесс, создавая специализированные приложения, доступные непосредственно в браузере. Однако большинство существующих решений либо требуют подключения к онлайн-сервисам, либо зависят от тяжёлых фреймворков, что усложняет их развёртывание и использование. В связи с этим возникает практическая необходимость в создании легковесных, автономных веб-приложений, работающих локально без внешних зависимостей.

Генеративные языковые модели открывают новую парадигму разработки, позволяя быстро создавать функциональные приложения на основе текстовых описаний требований. При этом важно понять границы применимости данного подхода: способен ли искусственный интеллект справиться с задачами, требующими точной работы с физическими размерами, автоматической раскладки элементов на печатном листе и соблюдения строгих полиграфических стандартов?

1.2 Обзор исследования

Исследования в области применения больших языковых моделей для генерации кода активно развиваются в последние годы. Работы С. В. Рогатина и В. А. Антошкина закладывают основы понимания возможностей нативных веб-технологий для создания интерактивных приложений без использования сторонних плагинов [1]. А. Б. Рахмонов исследует концепцию вайб-кодинга в образовательном контексте, подчёркивая его потенциал для демократизации процесса разработки [2].

И. В. Елманов предлагает использовать методологию Spec Driven Development для валидации кода, сгенерированного нейросетями, что подчёркивает важность строгого контроля качества при генеративной разработке [3]. Фундаментальные исследования А. Д. Степанова и коллег демонстрируют успешное применение ИИ-инструментов в областях, требующих высокой точности и работы со структурированными данными [5].

Однако специфика полиграфических задач – необходимость точного соблюдения физических размеров, работа с двусторонней печатью, автоматическая раскладка элементов – остаётся малоизученной в контексте генеративной разработки. Данное исследование призвано заполнить этот пробел.

1.3 Цель исследования

Цель работы заключается в оценке возможностей больших языковых моделей (на примере Qwen Studio) при создании специализированного веб-приложения для генерации кувертных карт с поддержкой нескольких форматов, автоматической раскладкой на печатные листы и соблюдением точных полиграфических размеров в рамках архитектурной парадигмы Zero-Dependency.

2 Материалы и методы

Исследование основано на подходе итеративной генерации кода с использованием большой языковой модели Qwen Studio. Основным инструментом выступал детализированный промпт-инжиниринг – процесс последовательного уточнения требований к функционалу, дизайну и техническим характеристикам приложения.

В качестве тестового проекта выбрана разработка веб-приложения для создания кувертных карт. Этот пример представляет практический интерес, так как сочетает несколько сложных аспектов:

- работу с точными физическими размерами (миллиметры)
- автоматическую раскладку элементов на листе формата A4 (210×297 мм)
- генерацию двусторонних развёрток с линией сгиба
- адаптивную типографику для различных форматов карт
- массовую обработку данных (списков имён гостей).

Эксперимент проводился в несколько этапов:

1. Первоначальная генерация – создание базовой версии приложения с одним форматом карт (195×110 мм).
2. Итеративное уточнение – последовательное добавление и корректировка функционала на основе тестирования.
3. Расширение функционала – добавление поддержки дополнительных форматов (маленькие 90×110 мм и большие 195×165 мм).

Ключевым ограничением стал полный отказ от внешних библиотек и фреймворков. Всё приложение должно размещаться в едином HTML-файле, содержащем только нативный HTML, CSS и Vanilla JavaScript. Это позволяет

оценить способность ИИ самостоятельно выстраивать архитектуру приложения и управлять сложной логикой без помощи готовых решений.

3 Результаты

В ходе эксперимента было создано полнофункциональное веб-приложение для генерации кувертных карт. Приложение поддерживает три формата карт и автоматически рассчитывает оптимальную раскладку на листах А4.

Первоначальный запрос к языковой модели включал требования к созданию приложения с классическим форматом карт 195×110 мм.

Промт для генерации: «Привет. Действуй как Senior Frontend разработчик. Твоя задача – написать полный, готовый к запуску код веб-приложения «Кувертные карты». Приложение предназначено для создания кувертных карт 195 на 165 миллиметров. Расположение карт: две карты на одном листе формата А4, стык в стык для удобства вырезания. Приложение должно быть реализовано в одном файле (HTML, CSS и JavaScript) и работать локально в браузере».

Результат генерации: <https://disk.yandex.ru/d/FLiw0nLnNIngXg>

В результате первой генерации кода получилось приложение без функционала как такового (рис. 1). С помощью него нельзя быстро создать и распечатать карты.

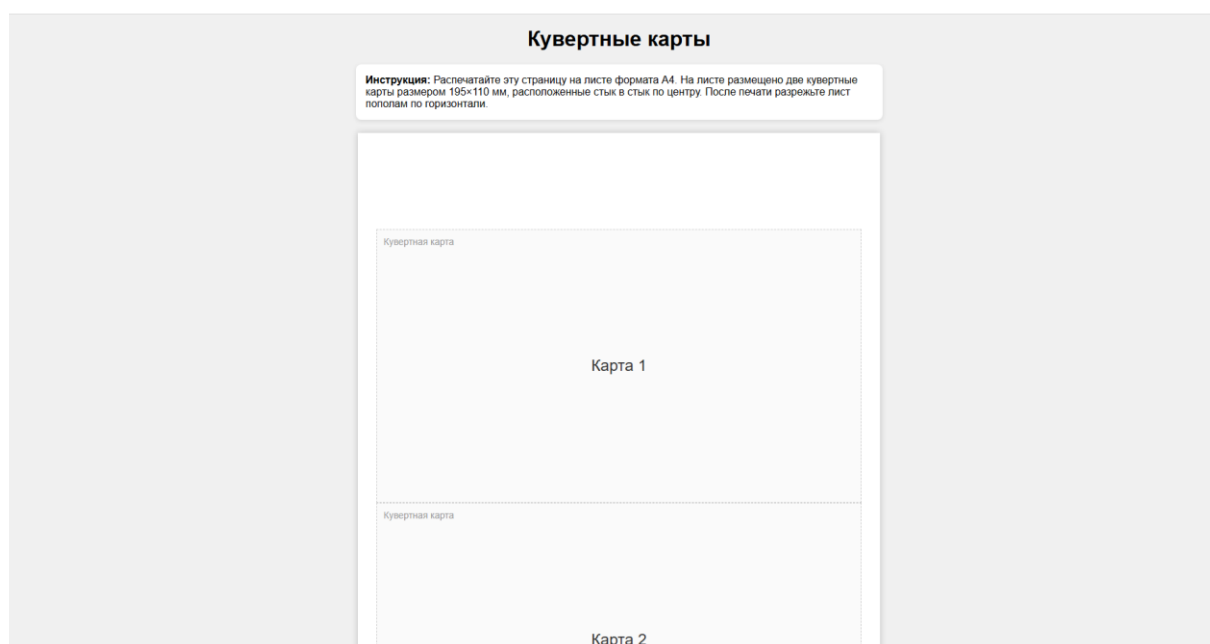


Рисунок 1. Результат первой генерации

Для исправления проблемы написан новый промт: «Исправь и добавь следующее:

1. Карты должны располагаться не по центру, а на краю листа для удобства вырезания
2. Добавь текстовое поле, в которое пишется фамилия, имя и отчество человека, для которого делается карта.

3. Сделай неограниченное количество карт, которые можно сделать (ФИО пишутся в текстовое поле, каждый с новой строки. Карты располагаются по две на листе).

4. Добавь кнопку «Создать карту», по нажатию которой карты создаются и выводится пример.

5. Добавь кнопку «Печать».

Результат генерации: <https://disk.yandex.ru/d/zmyANL-bMyrnvg>

Вторая версия приложения уже работала должным образом. Появилось текстовое поле, в которое пишутся имена гостей (рис. 2).

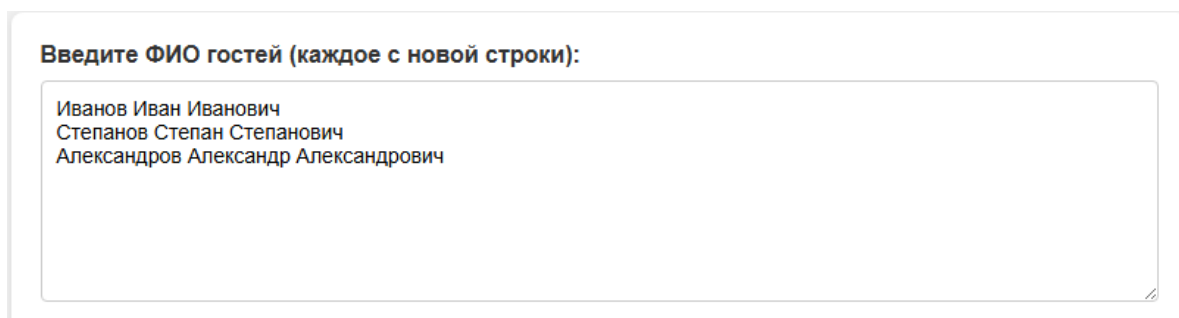
A screenshot of a web application interface. At the top, there is a label "Введите ФИО гостей (каждое с новой строки):". Below it is a large, empty text input area with a light gray border. Inside the input area, there is a small, faint text example: "Иванов Иван Иванович", "Степанов Степан Степанович", "Александров Александр Александрович".

Рисунок 2. Текстовое поле

Добавлены кнопки для просмотра и печати карт (рис. 3)

A screenshot of two buttons side-by-side. The left button is blue with the text "Создать карты" in white. The right button is green with the text "Печать" in white.

Рисунок 3. Кнопки

Также данный код реализует автоматическую разбивку списка на карты и размещение двух карт на одном листе А4. (рис. 4)

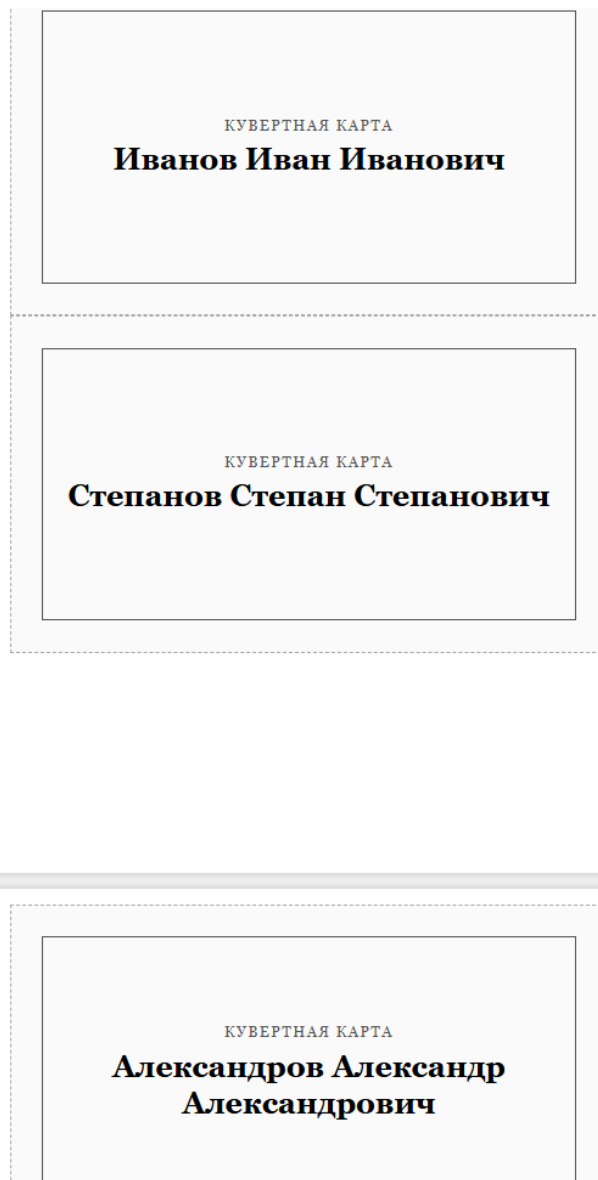


Рисунок 4. Примеры карт

На данном этапе приложение создаёт карточки с именами, но они всё ещё не являются кувертными, так как кувертная карта должна быть двусторонней для установки её в специальную рамку.

Для исправления данной ошибки написан следующий промт: «Исправь следующее:

1. Убери на самой карте надпись "Кувертная карта"
2. Карта должна быть двусторонней, чтобы её можно было согнуть, также добавь линию сгиба. ФИО дублируется с обеих сторон и располагается в рамке (одно из них перевёрнуто).
3. ФИО пишется в две строки. Фамилия заглавными буквами в первой строке и имя отчество с большой буквы во второй».

При написании промта также принято решение отформатировать расположение имени гостя.

Результат генерации: https://disk.yandex.ru/d/n4hv_bI1uiKBDw

При третьей генерации написан код, который реализовывает двустороннюю развёртку с линией сгиба посередине и форматирование ФИО (рис. 5).

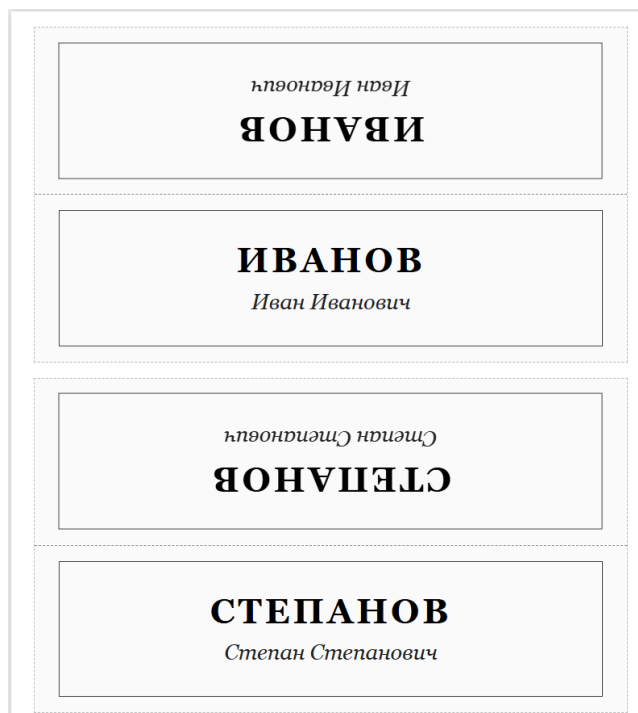


Рисунок 5. Готовые кувертные карты

На этом этапе разработки приложение готово к использованию. Все требования соблюдены, а кнопка «Печать» позволяет печатать неограниченное количество карт напрямую из браузера. Каждая карта состоит из двух половинок по 50% высоты с линией сгиба между ними. Верхняя половинка перевёрнута на 180°, что позволяет после сгиба получить карту, читаемую с обеих сторон. Однако принято решение расширить функционал и добавить поддержку нескольких форматов кувертных карт.

Промт для генерации: «Добавь возможность делать не только классические карты, но и большие и маленькие. Размеры: маленькие: 90 мм на 110 мм, большие: 195 мм на 165 мм. Большая карта располагается одна на лист. Расположение маленьких рассчитай сам с учётом размера листа и удобства для вырезания».

Результат генерации: https://disk.yandex.ru/d/p_FDSiupeNoBYg

Теперь приложение позволяет создавать кувертные карты не только классического размера (рис. 6-7). Например, такие карты применяются на официальных встречах в ресторанах.

Размер карт:

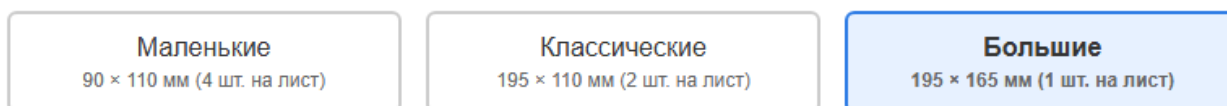


Рисунок 6. Окно выбора размера карт

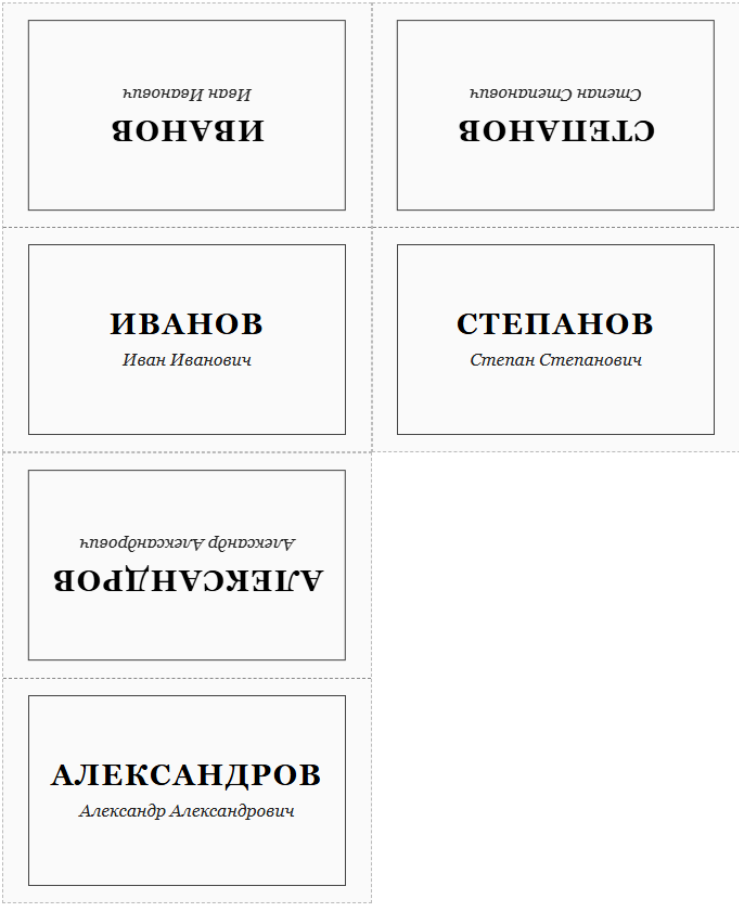


Рисунок 7. Маленькие кувертные карты

4 Вывод

Проведённое исследование продемонстрировало высокую эффективность больших языковых моделей при создании специализированных веб-приложений для полиграфических задач. Эксперимент по генерации приложения «Кувертные карты» в парадигме Zero-Dependency подтвердил несколько важных положений.

Искусственный интеллект успешно справился с задачами, требующими прецизионной работы с физическими размерами в миллиметрах, автоматического расчёта раскладки элементов на печатном листе и соблюдения пропорций.

Нейросеть реализовала систему адаптивных шрифтов, автоматически подбирающую размеры текста в зависимости от формата карты, что обеспечивает читаемость при сохранении эстетичности.

Приложение берёт на себя всю рутинную работу: разбиение списка имён на отдельные карты, расстановку их на листах, форматирование текста, подготовку к печати. Это позволяет пользователю сосредоточиться на содержании, а не на технической стороне.

Языковая модель спроектировала не просто функциональный код, а полноценный пользовательский интерфейс с визуальным переключателем форматов, подсказками, кнопками действий и превью результата.

Генеративная разработка с использованием больших языковых моделей представляет собой мощный инструмент для создания узкоспециализированных веб-приложений. В задачах, связанных с обработкой структурированных данных, соблюдением строгих правил и требований к дизайну, искусственный интеллект демонстрирует результаты, сопоставимые с работой профессиональных разработчиков. При этом скорость создания приложения увеличивается в разы, а порог входа для конечных пользователей снижается до минимума.

Библиографический список

1. Рогатин С. В., Антошкин В. А. Разработка компьютерных игр по технологии html5 //Информатика и прикладная математика. 2015. №. 21. С.12-16. URL: <https://elibrary.ru/item.asp?id=25030492>
2. Рахмонов А. Б. Вайб-кодинг как инструмент цифровой педагогики: функциональные возможности и механизмы реализации //The Lingua Spectrum. 2025. Т. 12. №. 2. С. 552-556. URL: <https://lingvospektr.uz/index.php/lngsp/article/view/1329>
3. Елманов И. В. Инженерный подход к ии: использование spec driven development для валидации vibe coding //Вестник науки. 2025. Т. 4. №. 12 (93). С. 716-723. URL: <https://cyberleninka.ru/article/n/inzhenernyy-podhod-kii-ispolzovanie-spec-driven-development-dlya-validatsii-vibe-coding>
4. Мухаметов Д. Р. Политический вайб-маркетинг: возможности искусственного интеллекта для продвижения образа будущего //Известия Тульского государственного университета. Гуманитарные науки. 2025. №. 3. С. 27-33. URL: <https://cyberleninka.ru/article/n/politicheskiy-vaybmarketing-vozmozhnosti-iskusstvennogo-intellekta-dlya-prodvizheniya-obrazabuduschego>
5. Степанов А. Д., Богданов Л. А., Шишкова Д. К., Кутихин А. Г. Применение инструментов на основе искусственного интеллекта в экспериментальной и трансляционной кардиологии: обзор //Российский кардиологический журнал. 2026. Т. 31. №. S2. С. 114-122. URL: <https://cyberleninka.ru/article/n/primenenie-instrumentov-na-osnoveiskusstvennogo-intellekta-v-eksperimentalnoy-i-translyatsionnoy-kardiologiiobzor>