

Проектирование интерактивных веб-приложений методами промпт-инжиниринга: разработка голосового блокнота в среде Qwen Studio

Пайдершев Алексей Константинович

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

Данная работа посвящена изучению потенциала больших языковых моделей в создании интерактивных веб-продуктов на примере генерации голосового блокнота с помощью платформы Qwen Studio. Использование встроенного браузерного интерфейса Web Speech API отвечает современным потребностям в быстрых и автономных решениях, способных обрабатывать речь локально, без подключения сторонних библиотек и внешних облачных серверов. Подобный подход значительно упрощает цикл разработки программного обеспечения: благодаря детализированному промпт-инжинирингу нейросеть успешно берет на себя функции разработчика. Вместе с тем, жесткие алгоритмические требования к потоковой обработке данных в реальном времени позволяют объективно оценить способность искусственного интеллекта писать сложный, выстраивать грамотную архитектуру приложения и обеспечивать мгновенный отклик интерфейса при строгом соблюдении стандартов браузерной безопасности.

Ключевые слова: Qwen Studio, нейросети, промпт-инжиниринг, Web Audio API, распознавание речи.

Designing interactive Web applications using prompt-engineering: developing a voice notepad in the Qwen Studio environment

Paydershev Alexey Konstantinovich

Sholom-Aleichem Priamursky State University

Student

Abstract

This work focuses on exploring the potential of large language models in creating interactive web products, using the Qwen Studio platform to generate a voice-activated notepad. The use of the built-in Web Speech API browser interface meets the current need for fast and autonomous solutions that can process speech locally, without the need for third-party libraries or external cloud servers. This approach significantly simplifies the software development cycle, as the neural network takes on the developer's responsibilities through detailed reverse engineering. At the same time, strict algorithmic requirements for real-time data streaming allow us to objectively assess the ability of artificial intelligence to write complex code, build a

competent application architecture, and ensure instant interface response while strictly adhering to browser security standards.

Keywords: Qwen Studio, neural networks, prompt-engineering, Web Audio API, speech recognition.

1 Введение

1.1 Актуальность

Актуальность данного исследования обусловлена стремительным развитием генеративных языковых моделей и их активным внедрением в процесс разработки программного обеспечения. Современная веб-индустрия зачастую перегружена многочисленными зависимостями и многоступенчатыми процессами сборки, что существенно замедляет создание стартовых прототипов и повышает порог входа для специалистов. В этих условиях возникает устойчивый практический запрос на поиск альтернативных, более быстрых и легковесных подходов к созданию функциональных цифровых продуктов.

Изучение методов структурированного промпт-инжиниринга в связке с максимальным использованием нативных браузерных технологий предлагает эффективное решение описанной проблемы. Переход к концепции автоматизированной генерации кода позволяет отказаться от избыточных сторонних сервисов, минимизировать рутинный труд и сместить фокус внимания на системное проектирование.

1.2 Обзор исследования

С. В. Рогатин и В. А. Антошкин описывают базовые принципы браузерной разработки, рассматривая технологию HTML5 как универсальную и надежную платформу для создания интерактивных приложений [1]. Концепцию современного взаимодействия с ИИ исследует А. Б. Рахмонов, который адаптирует этот метод для нужд цифровой педагогики и выделяет его функциональные возможности в образовательном процессе [2]. И. В. Елманов предлагает строгий инженерный подход к подобной генерации, обосновывая необходимость применения методологии Spec Driven Development для верификации и валидации результатов вайб-кодинга [3]. Выход генеративных технологий за рамки чистого программирования демонстрирует Д.Р.Мухаметов, анализирующий потенциал нейросетей и политического вайб-маркетинга для конструирования образа будущего [4]. А. Д. Степанов, Л. А. Богданов, Д. К. Шишкова и А. Г. Кутихин обобщают опыт применения ИИ-инструментов в точных науках, доказывая их высокую практическую значимость на примере решения сложных задач в экспериментальной и трансляционной кардиологии [5].

1.3 Цель исследования

Цель исследования – оценить эффективность структурированного промпт-инжиниринга для автоматизированной генерации функциональных веб-приложений на базе нативных браузерных API.

2 Материалы и методы

Методология исследования представляет собой гибридный подход, при котором традиционная веб-разработка заменяется автоматизированной генерацией кода посредством структурированного промпт-инжиниринга.

Особое внимание при составлении инструкции уделяется логике обработки данных в реальном времени. В запросе жестко фиксируются алгоритмические требования к потоковому распознаванию, включая обязательную поддержку непрерывной записи речи и динамический вывод промежуточных текстовых гипотез, что необходимо для создания эффекта мгновенного отклика интерфейса. Непосредственная генерация итогового кода, превращающего всю описанную логику в единый документ, осуществляется на базе платформы Qwen Studio, что позволяет создать сложный интерактивный инструмент с минимальным ручным вмешательством.

3 Результаты

Первый шаг исследования – написание промта для генерации кода.

Промт: Act as a Senior Frontend developer. Your task is to write a complete, ready-to-run code for the Voice Notepad (Speech-to-Text) web application. The application should be implemented in a single file and work directly in the browser.

Architectural requirements:

1. Use the «Web Speech API» built into the browser. Do not use third-party speech recognition libraries.

2. Enable support for continuous recognition and display of intermediate results, so that the user can see the text being typed in real time.

3. Set the default recognition language to Russian.

Interface requirements:

- Modern, minimalistic design. Color scheme: light with pleasant accent colors (e.g., blue or purple).

- There should be a large text field in the center of the screen where the recognized text will be recorded. The user should be able to edit it manually.

- Below the text field, there should be a control panel with the following buttons:

1. «Начать запись» (with a microphone icon, if possible, use emojis).

2. «Остановить».

3. «Копировать текст» (copies everything to the clipboard).

4. «Очистить».

- Visual status indicator: when recording is in progress, a pulsating red dot or the text «Идёт запись...» should appear on the screen.

Requirements for the code logic:

- Add a check for «Web Speech API» support in the browser. If the browser does not support this feature, display a nice error notification and block the buttons.

- The recognition logic should glue together the final results (when the person pauses) and the intermediate results (what is being said right now).

- Write clean, well-structured, and commented code.

Output the full code in one block so I can save it as index.html and run it immediately.) (Действуй как Senior Frontend разработчик. Твоя задача – написать полный, готовый к запуску код веб-приложения «Голосовой блокнот» (Speech-to-Text). Приложение должно быть реализовано в одном файле и работать прямо в браузере.

Архитектурные требования:

1. Используй встроенный в браузер «Web Speech API». Не используй сторонние библиотеки для распознавания речи.

2. Включи поддержку непрерывного распознавания и отображения промежуточных результатов, чтобы пользователь видел, как текст набирается в реальном времени.

3. Установи язык распознавания по умолчанию на русский.

Требования к интерфейсу:

- Современный, минималистичный дизайн. Цветовая гамма: светлая с приятными акцентными цветами (например, синий или фиолетовый).

- По центру экрана должно быть большое текстовое поле, куда будет записываться распознанный текст. Пользователь должен иметь возможность редактировать его вручную.

- Под текстовым полем панель управления с кнопками:

1. «Начать запись» (с иконкой микрофона, если возможно, используй эмоджи).

2. «Остановить».

3. «Копировать текст» (копирует всё в буфер обмена).

4. «Очистить».

- Визуальный индикатор статуса: когда идет запись, на экране должна появляться пульсирующая красная точка или текст «Идет запись...».

Требования к логике кода:

- Добавь проверку на поддержку «Web Speech API» в браузере. Если браузер не поддерживает эту функцию, выведи красивое уведомление об ошибке и заблокируй кнопки.

- Логика распознавания должна склеивать финальные результаты (когда человек сделал паузу) и промежуточные (то, что говорится прямо сейчас).

- Напиши чистый, хорошо структурированный и прокомментированный код.

Выведи полный код в одном блоке, чтобы я мог сохранить его как index.html и сразу запустить.)

Результат: <https://drive.google.com/file/d/1bUWZJCWMHL5lE6EJc-x6v2dU76SincrZ/view?usp=sharing>

В результате успешно сгенерирован полностью функциональный прототип веб-приложения «Голосовой блокнот». ИИ-ассистент сформировал готовый к запуску файл index.html, не требующий дополнительных правок или ручной отладки перед первым запуском.

Тестирование приложения подтвердило ожидаемые политики безопасности: браузер блокировал доступ к микрофону при попытке запуска

по протоколу <http://>, однако при локальном запуске (двойной клик по файлу `index.html`), прототип мгновенно получал доступ к оборудованию. Распознавание речи происходило практически без задержек, а функция копирования текста в буфер обмена работала корректно.

Интерфейс приложения интуитивно понятен пользователю, поэтому предварительного обучения перед использованием программы не нужно (рис 1).

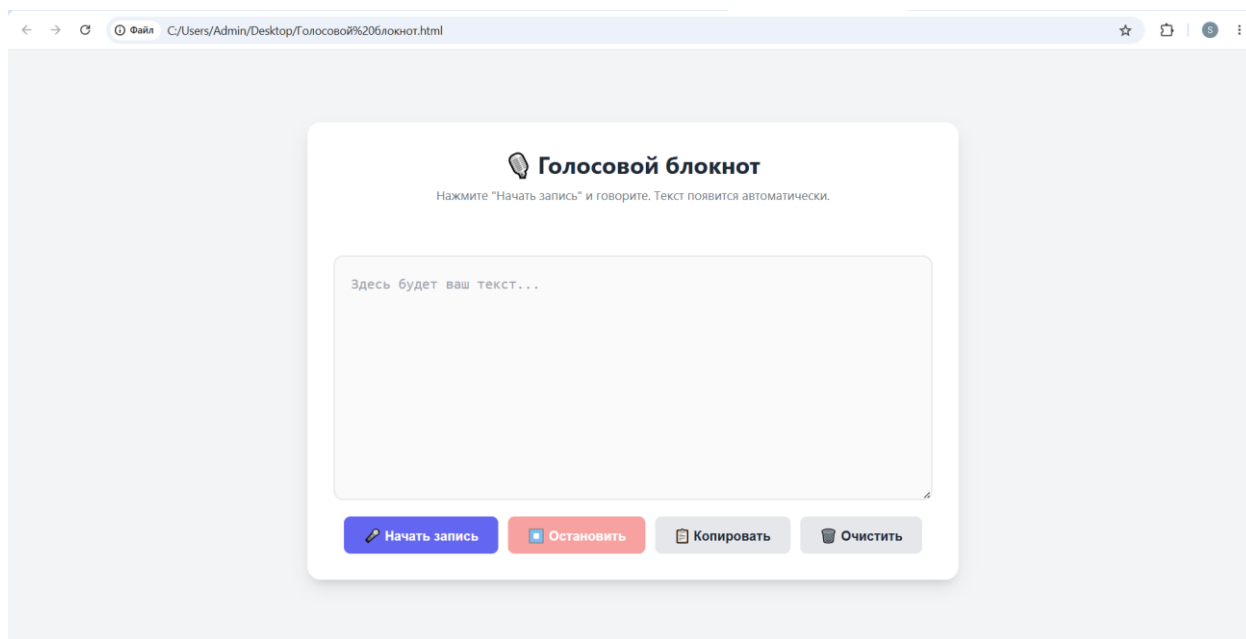


Рисунок 1. Интерфейс приложения

Созданное приложение обладает минимальными системными требованиями: браузер Google Chrome/Яндекс, микрофон и постоянный доступ к интернету.

Чтобы записать текст, необходимо нажать на кнопку «Начать запись». В верхней части появляется пульсирующая красная точка и надпись «Идет запись». Слова набираются в реальном времени – это промежуточные результаты. После небольшой паузы они становятся окончательными. Чтобы остановить запись, необходимо нажать на кнопку «Остановить».

Поле с текстом полностью доступно для редактирования. В нём можно исправлять опечатки, удалять лишнее или дописывать что-то вручную даже во время записи. Нажатие кнопки «Копировать» одним кликом копирует весь текст в буфер обмена. После нажатия кнопка на 2 секунды показывает сообщение «Скопировано!» (рис. 2).

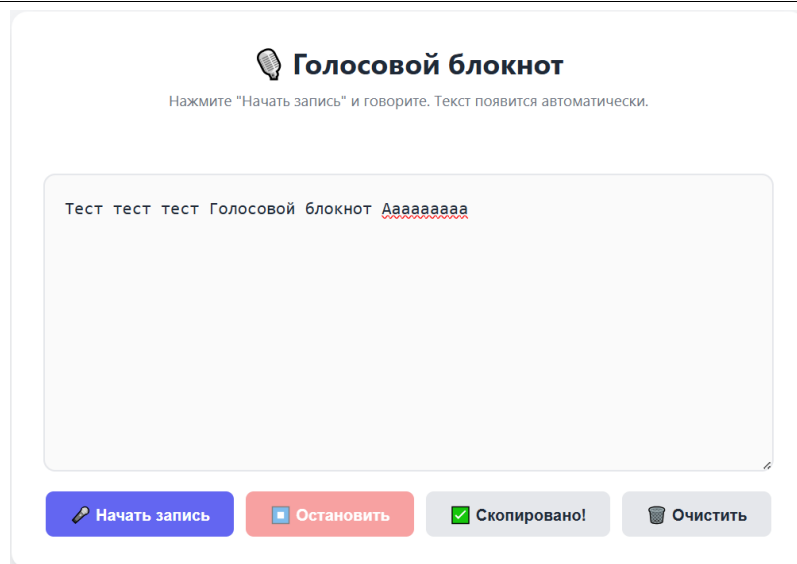


Рисунок 2. Уведомление о копировании

Кнопка «Очистить» удаляет весь текст из поля с текстом. Во избежание случайного удаления текста, появляется подтверждение об удалении, чтобы случайно не потерять запись (рис. 3).

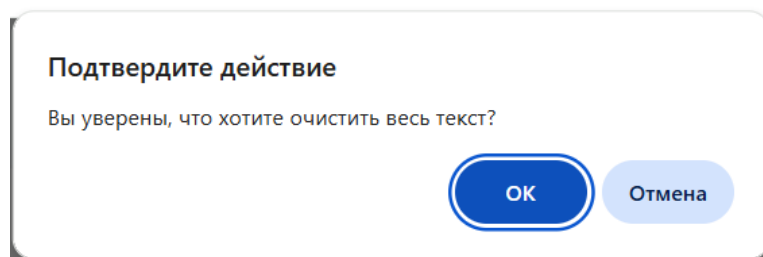


Рисунок 3. Подтверждение удаления текста

Web Speech API имеет особенность – он может самопроизвольно завершать сессию после нескольких секунд тишины. В коде реализована функция, которая автоматически перезапускает движок, если пользователь не нажимал кнопку «Остановить». Также он не добавляет точки и запятые автоматически, для этого нужно произносить знаки препинания вслух и расставлять их вручную.

Также соблюдены требования конфиденциальности: приложение работает локально и не сохраняет текст на сервер, аудио отправляется на сервера распознавания речи только на время записи и не хранится, после закрытия все данные исчезают.

Структурированный промпт-инжиниринг доказал свою эффективность. Детальное описание технического задания и параметров API позволило ИИ-модели выступить в роли полноценного Senior-разработчика, сократив время от идеи до рабочего прототипа до нескольких минут.

4 Выводы

Проведенное исследование подтверждает, что современные генеративные модели, такие как Qwen Studio, выступают эффективным инструментом для быстрого создания веб-приложений. Детализированный запрос со строгими архитектурными ограничениями позволил создать полностью функциональный «Голосовой блокнот» в рамках одного файла, исключая необходимость ручного написания базового кода и настройки серверов. Подобный результат наглядно демонстрирует, что сегодня на первый план выходит умение проектировать архитектуру и точно формулировать техническое задание для искусственного интеллекта, а не исключительно знания программирования.

Несмотря на то, что нейросеть успешно интегрирует нативные браузерные технологии, она не заменяет человека-оператора полностью. За разработчиком сохраняется контролирующая роль – необходимо учитывать ограничения программной среды, такие как строгие требования безопасности. Тем не менее, применение структурированного промпт-инжиниринга значительно снижает технический барьер и открывает новые перспективы для создания легковесных, автономных программ, работающих прямо в браузере пользователя.

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-k-ii-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-vayb-marketing-vozmozhnosti-iskusstvennogo-intellekta-dlya-prodvizheniya-obraza-buduschego>
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-osnove-iskusstvennogo-intellekta-v-eksperimentalnoy-i-translyatsionnoy-kardiologii-obzor>