

Получи список последних принятых PR <https://github.com/langchain-ai/langchain>, сформулируй CHANGELOG и из него забавный текст, что в нашем проекте нового. После чего сделай на основе этого текста страницу в стиле 90'ых



Рассуждения:

Необходимо получить список последних принятых Pull Request (PR) из репозитория `langchain-ai/langchain`, на основе этих данных сформулировать CHANGELOG, создать забавный текст о новых изменениях, а затем создать веб-страницу в стиле 90-х.

План:

1. Получить список последних принятых PR из репозитория `langchain-ai/langchain`.
2. Сформулировать CHANGELOG на основе полученных данных.
3. Создать забавный текст, основываясь на CHANGELOG.
4. Создать веб-страницу в стиле 90-х используя полученный текст.

Начать выполнение с первого шага: запросить список последних принятых PR.

Шаг 1: Получение списка последних принятых PR.

Действие: Получение списка PR (GitHub)

```
{"owner":"langchain-ai","repo":"langchain","state":"closed","sort":"updated","direction":"desc","per_page":10}
```

➤ Результат выполнения инструмента: Получение списка PR (GitHub)

Рассуждения:

У нас есть доступ к данным о последних PR, но они слишком длинные чтобы их сразу обрабатывать в одном фрагменте кода. Мне нужно извлечь информацию о PR, которые были приняты (merged), чтобы сформулировать CHANGELOG.

Действие: Код-интерпретатор

```
import json

# Загрузим данные о последних PR
data = json.loads('"<полученные_данные>"')

# Отфильтруем и выберем только те PR, которые были приняты
merged_prs = [pr for pr in data if pr['merged_at'] is not None]

# Сформируем CHANGELOG, извлекая номера и заголовки PR
changelog = [f"PR #{pr['number']}: {pr['title']}" for pr in merged_prs]
changelog
```

➤ Результат выполнения инструмента: Код-интерпретатор

Рассуждения:

Ошибка произошла из-за использования placeholder "`<полученные_данные>`", который не заменен на фактические данные. Мне нужно переопределить код для работы с переменной `function_results[0]['data']`.

Действие: Код-интерпретатор

```
import json

# Извлечем данные из переменной
data = function_results[0]['data']

# Отфильтруем и выберем только те PR, которые были приняты
merged_prs = [pr for pr in data if pr['merged_at'] is not None]

# Сформируем CHANGELOG, извлекая номера и заголовки PR
changelog = [f"PR #{pr['number']}: {pr['title']}" for pr in merged_prs]
changelog
```

➤ Результат выполнения инструмента: Код-интерпретатор

Рассуждения:

У меня есть CHANGELOG на основе полученных данных о принятых PR. Следующим шагом будет создание забавного текста на основе CHANGELOG.

CHANGELOG:

- PR #32556: chore: update integrations table
- PR #32555: docs: formatting Tavily
- PR #32553: docs: add pre-commit setup instructions to the dev setup guide
- PR #32530: fix(core): Support no-args tools by defaulting args to empty dict
- PR #32545: chore: add Chat LangChain to README.md
- PR #32552: docs: update banner
- PR #32550: docs: banner for academy course
- PR #32547: docs(ollama): update API usage examples
- PR #32540: docs: update outdated README.md content

Теперь нужно создать забавный текст на основе этих изменений. На этот раз у нас есть много новых интересных изменений!

🔧 Приветствуем наш интеграционный столик, который обрел новую жизнь с обновлениями, прям как превращение тыквы в карету!

📖 Встретили нового героя — Tavily! Его украшают новейшие форматирования, так что теперь он как король на страницах документации.

👉 Предприятием месяца стал «Как настроить pre-commit ?». Теперь это не только ваш личный пазл!

🔧 Наши инструменты стали еще более дружелюбны. Теперь они поддерживают пустое меню, просто полюбуйте на их доступную душу!

💬 Встречайте Chat LangChain в обновленном README.md. Теперь он как радио для тихих вечеров в компании с кодом.

🎨 Больше баннеров, больше курсов! Настало время веселых закладок и ярких заголовков!

💡 Всем пример — обновлённые примеры по использованию API! Теперь ещё светлее и ещё понятнее.

📖 Любимая 'старушка' README.md тоже изменилась, чтобы выглядеть как в молодые 90-е.

Переходим к созданию веб-страницы в стиле 90-х с использованием этого текста.

Продолжай



Рассуждения:

Теперь, когда у нас есть забавный текст, нужно создать веб-страницу в стиле 90-х с использованием этого текста. Для этого воспользуюсь инструментом `create_landing`.

План:

- 1. Использовать созданный забавный текст для создания веб-страницы.
- 2. Убедиться, что стиль страницы соответствует стилю 90-х.

Шаг 1: Создание веб-страницы в стиле 90-х.

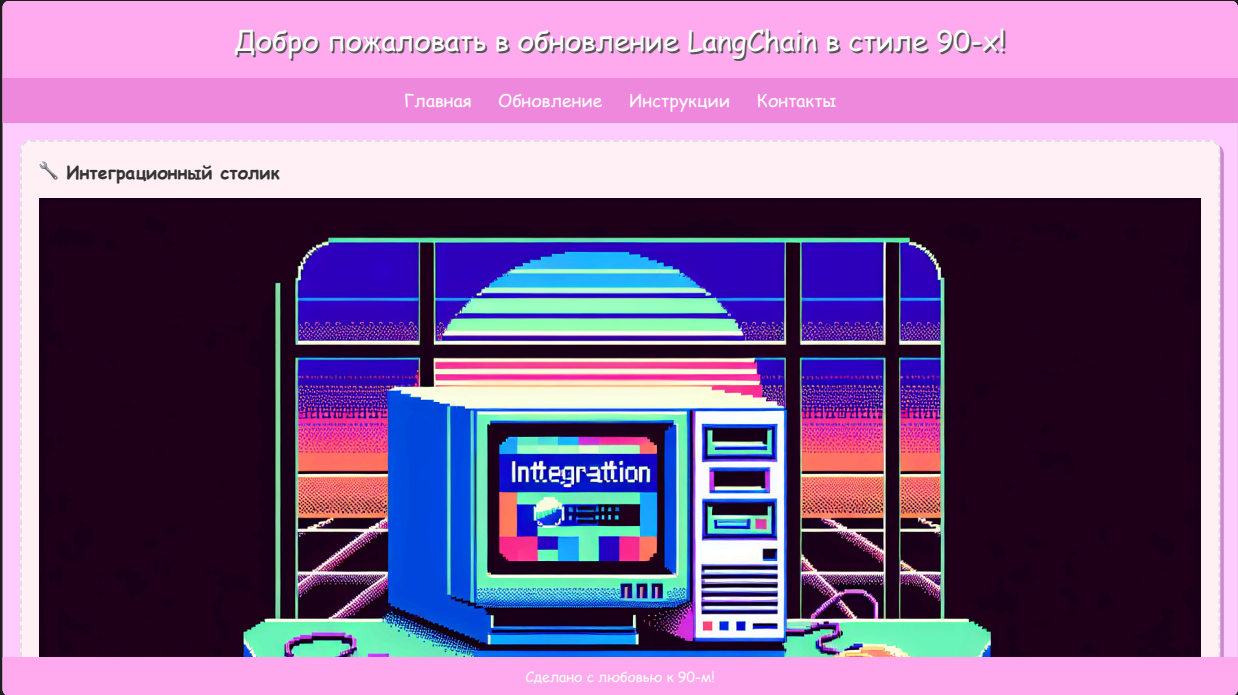
Действие: Создание веб-страницы

`{"task": "Создать веб-страницу в стиле 90-х с забавным текстом о новых изменениях в проекте LangChain. Те`

> Результат выполнения инструмента: Создание веб-страницы

В результате работы была сгенерирована HTML-страница `96fcf01f-e43e-4669-a2b4-ca4b03dfd8d3`

Веб-страница в стиле 90-х с забавным текстом о LangChain успешно создана! Страница включает в себя пиксель-арт изображения и стили, которые полностью соответствуют духу 90-х.



Если у вас есть дополнительные изменения или улучшения, которые вы хотите сделать на этой странице, дайте знать!

