

Техническое описание проекта: BannerAdsAI

1. Обзор проекта

BannerAdsAI — это веб-приложение, предназначенное для автоматического создания профессиональных рекламных баннеров. Пользователь вводит URL своего сайта, выбирает параметры, и искусственный интеллект генерирует релевантные заголовки и визуальные креативы.

Основная цель:

Максимально упростить и ускорить процесс создания рекламных материалов, сделав его доступным даже для пользователей без навыков дизайна.

2. Ключевая функциональность (Пользовательский сценарий)

Конфигурация (Шаг 1-3):

Пользователь вводит URL посадочной страницы.

Выбирает один из предустановленных размеров баннера (например, 300x250).

Выбирает цветовую схему (например, "синий фон + белый текст").

Генерация заголовков (Шаг 4):

На основе введенного URL и выбранных параметров, приложение обращается к AI-модели для анализа контента страницы.

Генерируются 3 уникальных заголовка в разных маркетинговых стилях (например: "прямая выгода", "решение проблемы", "призыв к действию").

Заголовки автоматически приводятся к верхнему регистру (КАПСЛОК).

Пользователь выбирает наиболее подходящий заголовок.

Генерация баннеров (Шаг 5):

Приложение генерирует 3 варианта фоновых изображений с помощью AI, основываясь на тематике сайта и выбранном заголовке.

На стороне клиента (в браузере) с помощью HTML5 Canvas на каждое сгенерированное изображение накладывается плашка с выбранным заголовком. Размер текста и высота плашки подбираются динамически для лучшей читаемости.

Пользователь видит готовые баннеры и может скачать их в формате .png.

3. Стек технологий

Проект построен на платформе base44, которая предоставляет готовую инфраструктуру (BaaS - Backend as a Service).

Фронтенд:

Язык: JavaScript (ES6+)

Фреймворк: React.js

Стилизация: Tailwind CSS

UI-компоненты: shadcn/ui (предоставляются платформой)

Анимации: framer-motion

Иконки: lucide-react

Рендеринг текста на изображениях: HTML5 Canvas API

Бэкенд и Платформа (base44):

База данных:

Встроенная NoSQL база данных base44 Entities. В проекте используется сущность BannerConfig для хранения конфигурации баннера.

AI-интеграции:

InvokeLLM: Для генерации текстового контента (заголовки, промпты для изображений). Аналог обращения к моделям вроде GPT-4.

GenerateImage: Для генерации фоновых изображений. Аналог обращения к моделям вроде DALL-E 3 или Midjourney.

Хостинг и деплой:

Управляется платформой base44.

4. Структура проекта

/

└─ entities/

└─ BannerConfig.json # JSON Schema для данных конфигурации баннера.

|

```
└─ pages/
  └─ BannerGenerator.jsx # Главный и единственный компонент страницы.
    # Управляет шагами, хранит основной стейт `config`.
    └─ components/
      └─ generator/
        └─ HeadlineStep.jsx # Компонент для Шага 4: генерация и выбор заголовков.
          # Содержит основной промпт для InvokeLLM.
          └─ BannerStep.jsx # Компонент для Шага 5: генерация и отображение баннеров.
            # Содержит логику генерации промпов для изображений
            # и функцию createBannerWithText для работы с Canvas.
└─ Layout.js # Глобальный компонент-обертка для всех страниц.
  # Содержит хедер, футер и глобальные CSS-стили.
```

5. Ключевая логика и поток данных

Состояние: Центральный объект состояния `config` находится в `BannerGenerator.jsx` и передается дочерним компонентам через пропсы.

Генерация заголовков (`HeadlineStep.jsx`):

Функция `generateHeadlines` формирует подробный промпт для `InvokeLLM`.

Промпт включает в себя URL сайта, требования к стилям, длине, использованию КАПСЛОКА и отсутствию кавычек.

Вызывается `await InvokeLLM({ prompt, add_context_from_internet: true })`.

Полученный текстовый ответ парсится, очищается от лишних символов и сохраняется в стейт.

Генерация баннеров (`BannerStep.jsx`):

Сначала вызывается `InvokeLLM` для создания качественного промпта для генерации изображения на английском языке.

Затем асинхронно (`Promise.all`) вызывается `GenerateImage` 3 раза для получения разных фонов.

Для каждого полученного URL изображения вызывается функция `createBannerWithText`:

Создается `<canvas>` нужного размера.

Рисуется фоновое изображение.

Рисуется цветная плашка (градиент).

Динамически подбирается размер шрифта, чтобы заголовок уместился в плашку.

Текст (в верхнем регистре) наносится на плашку с небольшой тенью для контраста.

`canvas` экспортируется в `blob`, создается временный URL (`URL.createObjectURL`) для отображения в `` и скачивания.