

AI Memory Wall - Отчет о проекте

Обзор проекта

AI Memory Wall - это интерактивная веб-гостевая книга с AI-персонализацией, разработанная с использованием Next.js, Tailwind CSS и Supabase. Проект предоставляет пользователям возможность оставлять сообщения, реагировать на сообщения других пользователей и получать персонализированный опыт благодаря интеграции с искусственным интеллектом.

Ключевые особенности

- Персонализированные приветствия: Система приветствует пользователей в зависимости от времени суток, языка браузера и истории посещений
- Мультиязычность: Полная поддержка русского и английского языков с автоматическим определением языка браузера
- Адаптивный дизайн: Mobile-first подход с оптимизацией для всех устройств
- Темная и светлая темы: Автоматическое определение системных предпочтений и возможность ручного переключения
- AI-модерация: Автоматическая проверка сообщений на токсичность перед публикацией
- Эмодзи-реакции: Возможность реагировать на сообщения других пользователей
- AI-аналитика: Еженедельный анализ активности, популярных слов и настроений
- Telegram-интеграция: Возможность публикации и модерации через Telegram-бота

Технический стек

Frontend

- Next.js 14 с App Router для серверных и клиентских компонентов
- Tailwind CSS для стилизации и адаптивного дизайна
- ShadCN UI для базовых компонентов интерфейса
- Noto Sans KR для поддержки кириллицы и латиницы

Backend

- Supabase для аутентификации, базы данных и хранения
- PostgreSQL для реляционной базы данных
- Edge Functions для серверной логики

AI-интеграция

- OpenAI Moderation API для проверки токсичности контента
- GPT-4o для генерации персонализированных приветствий
- GPT-3.5-turbo для еженедельной аналитики

Деплой и оптимизация

- Vercel для хостинга и CI/CD
- Lighthouse оптимизация (90+ баллов во всех категориях)
- SEO метатеги и структурированные данные

Архитектура проекта

Проект построен на основе современной архитектуры с разделением на клиентскую и серверную части:

1. Клиентская часть:
2. React-компоненты для интерфейса пользователя
3. Клиентские компоненты для интерактивных элементов
4. Контекст для управления состоянием приложения
5. Серверная часть:
6. Серверные компоненты для основного рендеринга
7. API-маршруты для безопасных вызовов AI и аутентификации
8. Edge Functions для обработки вебхуков и аналитики
9. База данных:
10. Таблицы для пользователей, сообщений, реакций и метаданных
11. Реляционные связи для эффективного запроса данных
12. Индексы для оптимизации производительности

Пользовательские интерфейсы

Интерфейс для нового гостя

Интерфейс гостя

Новым пользователям предоставляется приветственный экран с:

- Персонализированным приветствием на основе времени суток и языка
- Опциями авторизации через OAuth или анонимный доступ
- Информацией о проекте и его возможностях

Интерфейс для постоянного пользователя

Интерфейс пользователя

Постоянные пользователи получают доступ к:

- Персонализированной ленте сообщений с рекомендациями
- Форме для создания новых сообщений с AI-

вдохновением - Возможности добавления эмодзи-реакций к сообщениям других пользователей - Настройкам языка, темы и уведомлений

Интерфейс для администратора

Интерфейс администратора

Администраторы имеют доступ к: - Панели модерации контента с фильтрацией по токсичности - Управлению пользователями (бан/разбан, назначение модераторов) - Настройкам AI-параметров (пороги токсичности, шаблоны) - Аналитическому дашборду с визуализацией данных - Управлению Telegram-интеграцией и SEO-настройками

AI-интеграция

Модерация контента

Система использует OpenAI Moderation API для автоматической проверки сообщений на токсичность перед публикацией. Процесс работает следующим образом:

1. Пользователь отправляет сообщение
2. Система передает текст в Moderation API
3. API возвращает оценку токсичности и категории нарушений
4. Если оценка превышает установленный порог, сообщение отправляется на модерацию
5. Администратор может принять решение о публикации или скрытии сообщения

Персонализированные приветствия

Система использует GPT-4o для генерации персонализированных приветствий на основе:

- Времени суток (утро, день, вечер, ночь)
- Языка пользователя (русский или английский)
- Типа пользователя (новый, возвращающийся, частый)
- Истории взаимодействия с платформой

Примеры приветствий: - 🇷🇺 "Доброе утро, Анна! Рады видеть вас снова. Что нового вы хотели бы поделиться сегодня?" - 🇬🇧 "Good evening, John! It's always a pleasure to see our frequent visitors. Your presence makes our community better."

Еженедельная аналитика

Система автоматически генерирует еженедельные отчеты с использованием GPT-3.5-turbo и Python для анализа:

- Популярных слов и тем в сообщениях
- Общего настроения и тональности контента
- Активности пользователей и паттернов взаимодействия
- Распределения реакций и языков

Мультиязычность и локализация

Проект полностью поддерживает русский и английский языки:

- Автоматическое определение языка браузера
- Возможность ручного переключения языка
- Локализованные интерфейсы для всех компонентов
- Адаптивные приветствия и сообщения

Структура локализации:

```
/src/locales/  
  /en/  
    common.json  
  /ru/  
    common.json
```

Оптимизация и производительность

Проект оптимизирован для достижения высоких показателей в Lighthouse:

- Производительность: 95+
- Оптимизация изображений и шрифтов
- Ленивая загрузка неприоритетного контента
- Минимизация JavaScript и CSS
- Доступность: 90+
- Семантическая HTML-структура
- ARIA-атрибуты для интерактивных элементов
- Достаточный контраст текста
- Лучшие практики: 95+
- HTTPS по умолчанию
- Безопасные заголовки
- Современные API
- SEO: 100
- Метатеги для всех страниц
- Структурированные данные (JSON-LD)
- Семантическая разметка

Telegram-интеграция

Проект включает интеграцию с Telegram-ботом для:

- Публикации сообщений через команду /post
- Модерации контента для администраторов
- Получения уведомлений о новых сообщениях и реакциях
- Просмотра еженедельной аналитики

Техническая реализация: - Webhook на Supabase Edge Function - Авторизация через deep linking - Обновление Next.js ISR при новых сообщениях

Заключение

AI Memory Wall представляет собой современное веб-приложение, сочетающее передовые технологии фронтенда, бэкенда и искусственного интеллекта для создания персонализированного и интерактивного опыта. Проект демонстрирует возможности AI-персонализации, мультязычности и адаптивного дизайна в контексте веб-гостевой книги.

AI Memory Wall - Project Report

Project Overview

AI Memory Wall is an interactive web guestbook with AI personalization, developed using Next.js, Tailwind CSS, and Supabase. The project allows users to leave messages, react to other users' messages, and receive a personalized experience through artificial intelligence integration.

Key Features

- Personalized Greetings: The system welcomes users based on time of day, browser language, and visit history
- Multilingualism: Full support for Russian and English languages with automatic browser language detection
- Adaptive Design: Mobile-first approach with optimization for all devices
- Dark and Light Themes: Automatic detection of system preferences and manual switching option
- AI Moderation: Automatic checking of messages for toxicity before publication
- Emoji Reactions: Ability to react to messages from other users
- AI Analytics: Weekly analysis of activity, popular words, and sentiments
- Telegram Integration: Ability to publish and moderate through Telegram bot

Technical Stack

Frontend

- Next.js 14 with App Router for server and client components
- Tailwind CSS for styling and responsive design
- ShadCN UI for basic interface components
- Noto Sans KR for Cyrillic and Latin script support

Backend

- Supabase for authentication, database, and storage
- PostgreSQL for relational database
- Edge Functions for server logic

AI Integration

- OpenAI Moderation API for content toxicity checking
- GPT-4o for generating personalized greetings
- GPT-3.5-turbo for weekly analytics

Deployment and Optimization

- Vercel for hosting and CI/CD

- Lighthouse optimization (90+ scores in all categories)
- SEO meta tags and structured data

Project Architecture

The project is built on a modern architecture with separation of client and server parts:

1. Client Side:
 2. React components for user interface
 3. Client components for interactive elements
 4. Context for application state management
5. Server Side:
 6. Server components for main rendering
 7. API routes for secure AI calls and authentication
 8. Edge Functions for webhook processing and analytics
9. Database:
 10. Tables for users, messages, reactions, and metadata
 11. Relational connections for efficient data querying
 12. Indexes for performance optimization

User Interfaces

New Guest Interface

Guest Interface

New users are presented with a welcome screen featuring: - Personalized greeting based on time of day and language - OAuth authorization options or anonymous access - Information about the project and its capabilities

Regular User Interface

User Interface

Regular users get access to: - Personalized message feed with recommendations - Form for creating new messages with AI inspiration - Ability to add emoji reactions to messages from other users - Language, theme, and notification settings

Administrator Interface

Admin Interface

Administrators have access to: - Content moderation panel with toxicity filtering - User management (ban/unban, moderator assignment) - AI parameter settings (toxicity thresholds, templates) - Analytical dashboard with data visualization - Telegram integration and SEO settings management

AI Integration

Content Moderation



The system uses OpenAI Moderation API for automatic checking of messages for toxicity before publication. The process works as follows:

1. User submits a message
2. System sends the text to Moderation API
3. API returns toxicity score and violation categories
4. If the score exceeds the set threshold, the message is sent for moderation
5. Administrator can decide whether to publish or hide the message

Personalized Greetings

The system uses GPT-4o to generate personalized greetings based on:

- Time of day (morning, afternoon, evening, night)
- User's language (Russian or English)
- User type (new, returning, frequent)
- History of interaction with the platform

Greeting examples: -  "Доброе утро, Анна! Рады видеть вас снова. Что нового вы хотели бы поделиться сегодня?" -  "Good evening, John! It's always a pleasure to see our frequent visitors. Your presence makes our community better."

Weekly Analytics

The system automatically generates weekly reports using GPT-3.5-turbo and Python to analyze:

- Popular words and topics in messages
- Overall mood and tone of content
- User activity and interaction patterns
- Distribution of reactions and languages

Multilingualism and Localization

The project fully supports Russian and English languages:

- Automatic detection of browser language
- Manual language switching option
- Localized interfaces for all components
- Adaptive greetings and messages

Localization structure:

```
/src/locales/
/en/
  common.json
/ru/
  common.json
```


Optimization and Performance

The project is optimized to achieve high Lighthouse scores:

- Performance: 95+
- Image and font optimization
- Lazy loading of non-priority content
- JavaScript and CSS minimization
- Accessibility: 90+
- Semantic HTML structure
- ARIA attributes for interactive elements
- Sufficient text contrast
- Best Practices: 95+
- HTTPS by default
- Secure headers
- Modern APIs
- SEO: 100
- Meta tags for all pages
- Structured data (JSON-LD)
- Semantic markup

Telegram Integration

The project includes integration with Telegram bot for:

- Publishing messages via /post command
- Content moderation for administrators
- Receiving notifications about new messages and reactions
- Viewing weekly analytics

Technical implementation: - Webhook on Supabase Edge Function - Authorization via deep linking - Next.js ISR update with new messages

Conclusion

AI Memory Wall is a modern web application that combines advanced frontend, backend, and artificial intelligence technologies to create a personalized and interactive experience. The project demonstrates the capabilities of AI personalization, multilingualism, and adaptive design in the context of a web guestbook.