





react_microservices_platform






Отчет о создании React-приложения для интеграции микросервисов платформы услуг

Выполненные задачи

1. Архитектура и настройка проекта

-  Инициализировано React приложение с TypeScript, TailwindCSS и Vite
-  Настроена современная архитектура с модульным разделением
-  Добавлены необходимые зависимости (Zustand, Axios, React Hook Form, Zod, React Hot Toast)
-  Создана система типов TypeScript для всех API интерфейсов

2. API интеграция с микросервисами

-  **outsourcing-auth** (Go, :8080) - API клиент для авторизации и управления услугами
-  **Search+Parameters** (Go, :8070) - API клиент для поиска и фильтрации
-  **BD Client-Company** (Python, :5000) - API клиент для БД клиентов/компаний
-  **DS-AM** (Go, :9003) - API клиент для обработки фотографий
-  Реализована система fallback на mock данные при недоступности API

- ☒ Настроена обработка ошибок и retry логика

3. Функциональные компоненты

- ☒ Главная страница с поиском и категориями услуг
- ☒ Поиск и фильтрация услуг с продвинутыми фильтрами
- ☒ Авторизация - модальное окно входа и страница регистрации
- ☒ Регистрация для клиентов и компаний с валидацией форм
- ☒ Навигация с адаптивным header и footer
- ☒ Карточки услуг с рейтингами, ценами и изображениями

4. State Management и архитектура




- ☒ Zustand store для управления аутентификацией
- ☒ Zustand store для управления услугами и поиском
- ☒ JWT токены с безопасным хранением в cookies
- ☒ Reactive состояние с автоматическими обновлениями

5. UI/UX и дизайн

- ☒ Современный responsive дизайн на TailwindCSS
- ☒ Использование цветовой схемы из существующих макетов (#006FEE)
- ☒ Radix UI компоненты для консистентного интерфейса
- ☒ Lucide React иконки
- ☒ Toast уведомления для пользователя
- ☒ Адаптивность для мобильных устройств

6. Тестирование и оптимизация

- ☒ Полное тестирование функциональности в dev режиме
- ☒ Production build и деплой

-  Browser-тестирование всех основных функций
-  Проверка отсутствия console ошибок
-  Тестирование адаптивности

Технологический стек

Frontend

- **React 18.3** + TypeScript
- **Vite** (build tool)
- **TailwindCSS** (стилизация)
- **Radix UI** (компоненты)
- **React Router** (маршрутизация)
- **Zustand** (state management)
- **React Hook Form + Zod** (формы и валидация)
- **Axios** (HTTP клиент)
- **React Hot Toast** (уведомления)

Архитектурные решения

- Single Page Application (SPA)
- Модульная архитектура с разделением слоев
- API абстракция для микросервисов
- TypeScript для типобезопасности
- Graceful fallback на mock данные

Ключевые достижения

1. Успешная интеграция микросервисов

Создана абстракция для работы с 4 различными микросервисами на разных технологиях (Go, Python), с обработкой ошибок и fallback механизмами.

2. Современный пользовательский интерфейс

Реализован clean, responsive UI следующий современным принципам UX с использованием цветовой схемы и стилей из существующих макетов.

3. Готовность к production

Приложение полностью готово к использованию, протестировано и задеплоено с рабочим URL.






4. Масштабируемая архитектура


Код организован модульно, что позволяет легко добавлять новые функции и микросервисы.

Демонстрация

Рабочий URL: <https://x3ysm28u6l.space.minimax.io>

Протестированные функции:

-  Загрузка главной страницы
-  Поиск услуг с фильтрацией
-  Просмотр категорий услуг
-  Регистрация пользователей
-  Адаптивность дизайна

-  Отсутствие ошибок в консоли

Дальнейшее развитие

Архитектура готова для расширения:

- Добавление системы сообщений
- Интеграция платежной системы
- Расширенные профили пользователей
- Система отзывов и рейтингов
- Push уведомления
- Мобильное приложение на базе того же API

Key Files

- `/workspace/outsourcing-platform/src/App.tsx`: Главный компонент приложения с роутингом и защищенными маршрутами
- `/workspace/outsourcing-platform/src/pages/HomePage.tsx`: Главная страница с поиском, категориями и рекомендуемыми услугами
- `/workspace/outsourcing-platform/src/pages/SearchPage.tsx`: Страница поиска услуг с фильтрами и результатами
- `/workspace/outsourcing-platform/src/pages/auth/LoginPage.tsx`: Страница авторизации пользователей
- `/workspace/outsourcing-platform/src/pages/auth/RegisterPage.tsx`: Страница регистрации для клиентов и компаний
- `/workspace/outsourcing-platform/src/lib/api/config.ts`: Конфигурация API клиентов для всех микросервисов
- `/workspace/outsourcing-platform/src/lib/api/auth.ts`: API клиент для сервиса авторизации (Go)
- `/workspace/outsourcing-platform/src/lib/api/search.ts`: API клиент для поискового сервиса (Go) с mock fallback
- `/workspace/outsourcing-platform/src/store/authStore.ts`: Zustand store для управления авторизацией

- `/workspace/outsourcing-platform/src/store/servicesStore.ts`: Zustand store для управления услугами и поиском
- `/workspace/outsourcing-platform/src/types/api.ts`: TypeScript типы для всех API интерфейсов
- `/workspace/outsourcing-platform/public/data/mock-services.json`: Mock данные для демонстрации функциональности
- `/workspace/outsourcing-platform/package.json`: Конфигурация проекта с зависимостями
- `/workspace/outsourcing-platform/PROJECT_README.md`: Документация проекта с описанием архитектуры и функциональности