

Lynkr - Детальный Рoadмап Проекта

Обзор Проекта

Lynkr — минималистичное приложение для сохранения и организации ссылок. "Память для интернета" с гибкими коллекциями и простым интерфейсом.

Технологический Стек

- **Frontend:** Nuxt 3, TypeScript
 - **Backend:** Supabase (PostgreSQL, Auth, Storage)
 - **UI Kit:** Shadcn (через UnoCSS или Windi + Tailwind)
 - **Анимации:** Motion Vue / Framer Motion for Vue
 - **Auth:** Supabase Auth
 - **Хостинг:** Vercel / Supabase Hosting
 - **База:** PostgreSQL JSONB для кастомных полей
-

Фазы Разработки

Фаза 1: Фундамент и Авторизация (2-3 недели)

Неделя 1-2: Настройка Проекта

Цели:

- Настроить основную инфраструктуру
- Реализовать систему авторизации
- Создать базовую архитектуру приложения

Задачи:

1. Инициализация проекта

- Создать Nuxt 3 проект с TypeScript
- Настроить Supabase клиент
- Конфигурировать Shadcn UI + UnoCSS/Tailwind
- Настроить ESLint, Prettier, Husky

2. Дизайн-система

- Создать color palette (ч/б стиль с акцентами)
- Определить типографику
- Создать базовые компоненты (Button, Input, Card)

- Настроить dark/light themes

3. База данных

- Спроектировать схему БД в Supabase
- Создать таблицы: users, collections, links, tags
- Настроить RLS (Row Level Security)
- Написать начальные SQL функции

4. Система авторизации

- Реализовать регистрацию/авторизацию
- Настроить OAuth (Google, GitHub)
- Создать middleware для защищенных роутов
- Реализовать управление состоянием пользователя

Результат: Работающая авторизация, базовая дизайн-система, настроенная БД

Фаза 2: Коллекции и Управление (2-3 недели)

Неделя 3-4: Коллекции

Цели:

- Создать систему коллекций
- Реализовать иерархию и вложенность
- Добавить drag & drop функционал

Задачи:

1. CRUD для коллекций

- Создание/редактирование/удаление коллекций
- Валидация данных
- API endpoints в Supabase

2. Иерархия коллекций

- Вложенные коллекции (parent-child)
- Breadcrumbs навигация
- Древовидная структура в сайдбаре

3. Кастомизация коллекций

- Цветовая маркировка
- Иконки для коллекций
- Переименование и описания

4. Drag & Drop

- Перемещение коллекций
- Изменение порядка
- Визуальные индикаторы при перетаскивании

Результат: Полнофункциональная система коллекций с иерархией

Фаза 3: Карточки и Ссылки (2-3 недели)

Неделя 5-6: Управление Ссылками

Цели:

- Реализовать систему карточек
- Добавить функционал сохранения ссылок
- Создать preview и metadata

Задачи:

1. CRUD для карточек

- Добавление ссылок (URL, заголовок, описание)
- Редактирование и удаление
- Перемещение между коллекциями

2. Metadata и Preview

- Автоматическое получение заголовка страницы
- Генерация превью изображений
- Фавиконы сайтов
- Кеширование metadata

3. Кастомизация карточек

- Цветовая маркировка
- Система избранного
- Кастомные заголовки и описания
- Поддержка markdown (опционально)

4. Порядок и позиционирование

- Drag & drop для карточек
- Кастомный порядок внутри коллекций
- Сохранение позиций

Результат: Полнофункциональная система карточек с metadata

Фаза 4: Теги и Группировки (1-2 недели)

Неделя 7-8: Система Тегов

Цели:

- Реализовать систему тегов
- Создать группировки и фильтры
- Добавить визуальные индикаторы

Задачи:

1. Система тегов

- Создание/редактирование тегов
- Автокомплит при добавлении
- Цветовая кодировка тегов

2. Группировки

- Группировка по тегам
- Группировка по дате
- Группировка по коллекциям
- Визуальные разделители

3. Фильтрация

- Фильтр по тегам
- Фильтр по дате добавления
- Фильтр по избранному
- Комбинированные фильтры

Результат: Гибкая система тегов и группировок

Фаза 5: Поиск и Фильтры (1-2 недели)

Неделя 9: Поиск

Цели:

- Реализовать полнотекстовый поиск
- Создать продвинутые фильтры
- Оптимизировать производительность

Задачи:

1. Поиск

- Полнотекстовый поиск по заголовкам и описаниям
- Поиск по тегам
- Поиск по URL
- Instant search с debounce

2. Продвинутые фильтры

- Фильтр по дате (диапазон)
- Фильтр по коллекциям
- Комбинированные фильтры
- Сохранение состояния фильтров

3. Оптимизация

- Индексы в PostgreSQL
- Виртуализация длинных списков
- Lazy loading карточек

Результат: Быстрый и эффективный поиск

Фаза 6: UI/UX и Анимации (2-3 недели)

Неделя 10-11: Финальная Стилизация

Цели:

- Довести дизайн до совершенства
- Добавить плавные анимации
- Оптимизировать под мобильные устройства

Задачи:

1. Дизайн-система

- Финализировать цветовую схему
- Создать консистентные компоненты
- Темная/светлая тема
- Типографика в стиле манги

2. Анимации

- Переходы между страницами
- Hover эффекты
- Micro-interactions

- Loading states
- Drag & drop анимации

3. Адаптивность

- Mobile-first подход
- Адаптивная сетка
- Жесты для мобильных
- Touch-friendly интерфейс

4. Доступность

- Keyboard navigation
- Screen reader support
- Color contrast
- Focus indicators

Результат: Отполированный, красивый и доступный интерфейс

Фаза 7: Деплой и Тестирование (1 неделя)

Неделя 12: Запуск

Цели:

- Подготовить приложение к продакшену
- Провести тестирование
- Запустить MVP

Задачи:

1. Подготовка к продакшену

- Оптимизация бандла
- SEO оптимизация
- Мониторинг ошибок
- Аналитика

2. Тестирование

- Unit тесты для критических функций
- E2E тесты основных сценариев
- Тестирование производительности
- Кроссбраузерное тестирование

3. Деплой

- Настройка CI/CD
- Деплой на Vercel/Supabase
- Настройка домена
- SSL сертификаты

4. Пользовательские тесты

- Альфа-тестирование с близкими друзьями
- Сбор обратной связи
- Итерации по результатам

Результат: Запущенное MVP приложение

Пост-MVP Развитие

Фаза 8: Расширенный Функционал

- **Права доступа:** Шаринг коллекций, публичные ссылки
- **Браузерное расширение:** Быстрое сохранение ссылок
- **PWA:** Mobile app опыт
- **API для разработчиков:** Lynkr API
- **AI интеграция:** Автоматические summaries ссылок

Фаза 9: Интеграции

- **Импорт данных:** Из браузера, Pocket, Readwise
- **Экспорт данных:** JSON, CSV, HTML
- **Webhook'и:** Интеграция с Zapier, IFTTT
- **Социальные функции:** Публичные коллекции, подписки

Фаза 10: Аналитика и Оптимизация

- **Пользовательская аналитика:** Как используют приложение
 - **Рекомендации:** AI-powered предложения
 - **Производительность:** Оптимизация для больших коллекций
 - **Монетизация:** Premium функции
-

Оценка Времени и Ресурсов

Временные Рамки

- **MVP:** 12 недель (3 месяца)

- **Пост-MVP:** 6-12 месяцев
- **Команда:** 1-2 разработчика + дизайнер (по желанию)

Ключевые Риски

1. **Сложность drag & drop:** Может потребовать больше времени
2. **Производительность:** Оптимизация для больших объемов данных
3. **Мобильная адаптация:** Сложность жестов и интерфейса

Критерии Успеха MVP

- ☒ Пользователь может создать аккаунт
- ☒ Создавать коллекции и вложенные коллекции
- ☒ Добавлять и редактировать ссылки
- ☒ Использовать теги и фильтры
- ☒ Находить ссылки через поиск
- ☒ Приложение работает на мобильных устройствах
- ☒ Красивый и интуитивный интерфейс

Техническая Архитектура

Структура Проекта

```
lynkr/  
├── components/      # Vue компоненты  
│   ├── ui/         # Базовые UI компоненты  
│   ├── collections/ # Компоненты коллекций  
│   ├── links/       # Компоненты ссылок  
│   └── common/      # Общие компоненты  
├── pages/           # Страницы приложения  
├── composables/     # Vue composables  
├── stores/           # Pinia stores  
├── utils/            # Утилиты  
├── types/            # TypeScript типы  
└── assets/           # Статические файлы
```

База Данных

```
sql
```


-- Основные таблицы

users (id, email, name, settings)

collections (id, user_id, name, color, icon, parent_id, position)

links (id, user_id, collection_id, url, title, description, color, position, metadata)

tags (id, user_id, name, color)

link_tags (link_id, tag_id)

Этот роадмап обеспечивает структурированный подход к разработке Lynkr, с четкими этапами и конкретными результатами на каждом этапе.