



Что мы знаем о ПВА?



Клей ПВА (поливинилацетат)

Дисперсия
в воде с
специ
Обл
сл



Что мы знаем о PWA?

- Сайты с доступом к графическому ускорителю.
- Поддержка офлайн-режима.
- Возможность кэширования.
- Достаточно простая установка под различные систему и аппаратные решения.
- Лёгкий доступ к функциям и информации при минимальных затратах на разработку и поддержку.

Хорошие стороны

- PWA — это гибкость системы.
- Интеграция с существующими системами.
- Единый доступ к необходимым инструментам и данным.

В каждой бочке...

- Необходимость адаптации устаревших систем в режиме PWA.
- Проблемы с безопасностью и конфиденциальностью данных.
- Может требовать дополнительного обучения и поддержки.

Жизненные циклы

- Десктоп: установка пакета и обновление.
- Телефоны: установка из магазина приложений и обновление.
- Веб-приложения: установка на клиенте, обновление на сервере.

Небольшая история справка...

- Концепция веб-приложений — Microsoft, 2000
- Apple — первые HTML-приложения, 2007
- Google Chrome — Service Worker, 2015
- Поддержка PWA — iOS, Safari и Edge, 2018
- Поддержка офлайн-режима — все популярные ОС и браузеры, 2020

А кому оно надо?

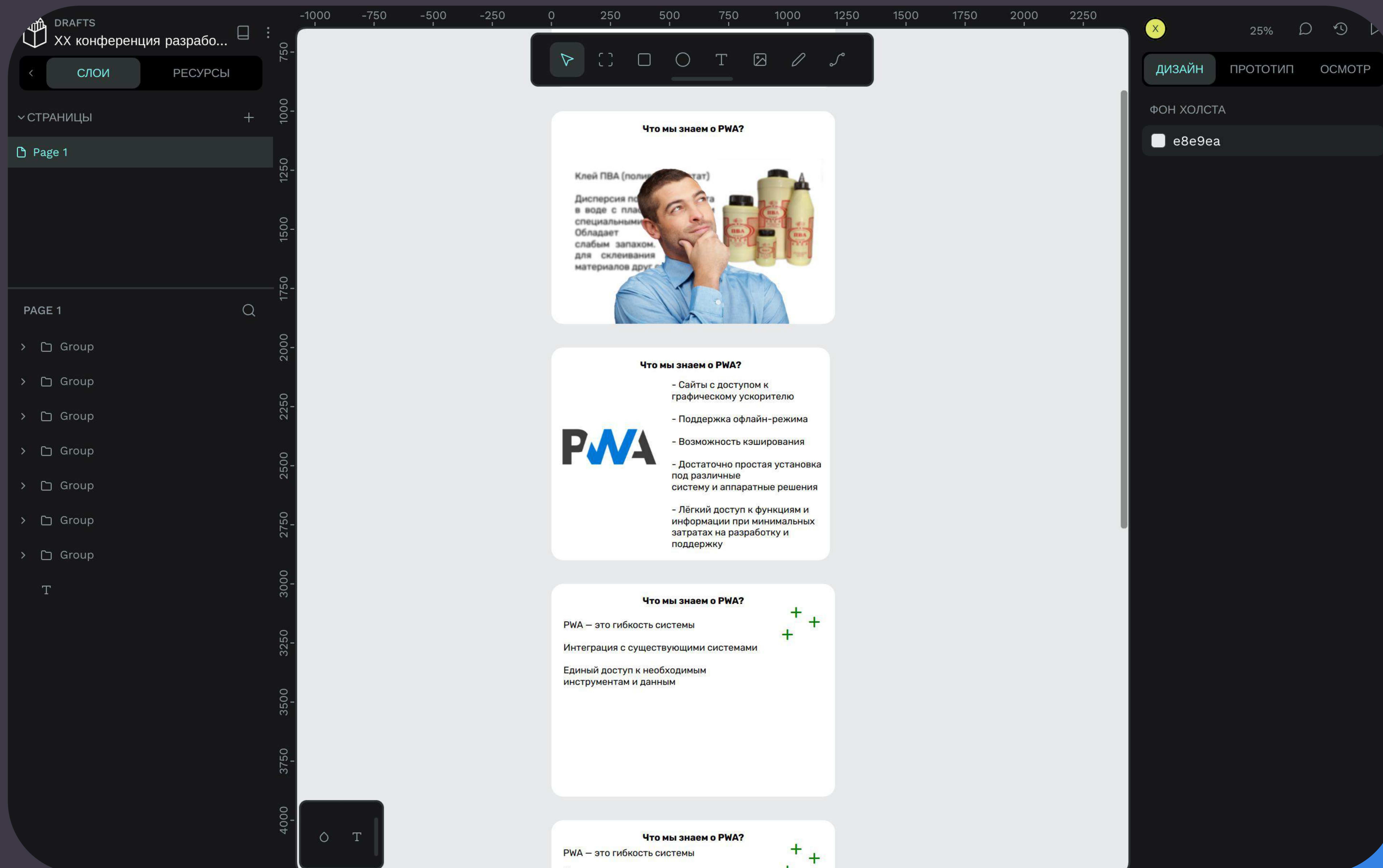
- Временное решение для бизнеса (первые 3 года)
- Устройства с малым объёмом ПЗУ
- Мобильные системы
- Работа в разных экосистемах (ALT Mobile, Android)

Что можно вернуть в PWA?

Всё*

* в рамках открытых сайтов, которые предполагают офлайн-режим использования, доступ к аппаратному ускорению и кэшированию, потому что данная технология крайне хорошо показывает себя на устройствах

Что можно завернуть в PWA?



Разработка своего PWA

- Старый сайт на NextJS и React
- PWAизация в одну команду, один файл и пару строк
- `npm install next-pwa`

ЕЩЁ НЕМНОГО...

public/manifest.json

```
{
  "name": "X1Z53 Website",
  "short_name": "X1Z53",
  "start_url": "/",
  "display": "standalone",
  "icons": [
    {
      "src": "/x1z53.svg",
      "sizes": "any",
      "type": "image/svg+xml"
    }
  ]
}
```

Ещё немного...

next.config.js

```
const withPWA = require("next-pwa")({  
  dest: "public",  
  register: true,  
  skipWaiting: true,  
})
```

```
module.exports = withPWA({  
  ...  
})
```

Сейчас всё ЕЩЁ проще

NEXT.js Showcase Docs Blog Templates Enterprise Search documentation... CtrlK Feedback Learn

Using App Router
Features available in /app

Getting Started
Installation
Project Structure

Building Your Application
Routing
Data Fetching
Rendering
Caching
Styling
Optimizing
Configuring

TypeScript
ESLint
Environment Variables
Absolute Imports and Module Path Aliases
MDX
src Directory
Custom Server
Draft Mode
Content Security Policy

App Router > ... > Configuring > Progressive Web Applications (PWA)

Progressive Web Applications (PWA)

Progressive Web Applications (PWAs) offer the reach and accessibility of web applications combined with the features and user experience of native mobile apps. With Next.js, you can create PWAs that provide a seamless, app-like experience across all platforms without the need for multiple codebases or app store approvals.

PWAs allow you to:

- Deploy updates instantly without waiting for app store approval
- Create cross-platform applications with a single codebase
- Provide native-like features such as home screen installation and push notifications

Creating a PWA with Next.js

1. Creating the Web App Manifest

Next.js provides built-in support for creating a [web app manifest](#) using the App Router. You can create either a static or dynamic manifest file:

For example, create a `app/manifest.ts` or `app/manifest.json` file:

```
app/manifest.ts TypeScript
```

On this page

Creating a PWA with Next.js

1. Creating the Web App Manifest
2. Implementing Web Push Notifications
3. Implementing Server Actions
4. Generating VAPID Keys
5. Creating a Service Worker
6. Adding to Home Screen
7. Testing Locally
8. Securing your application

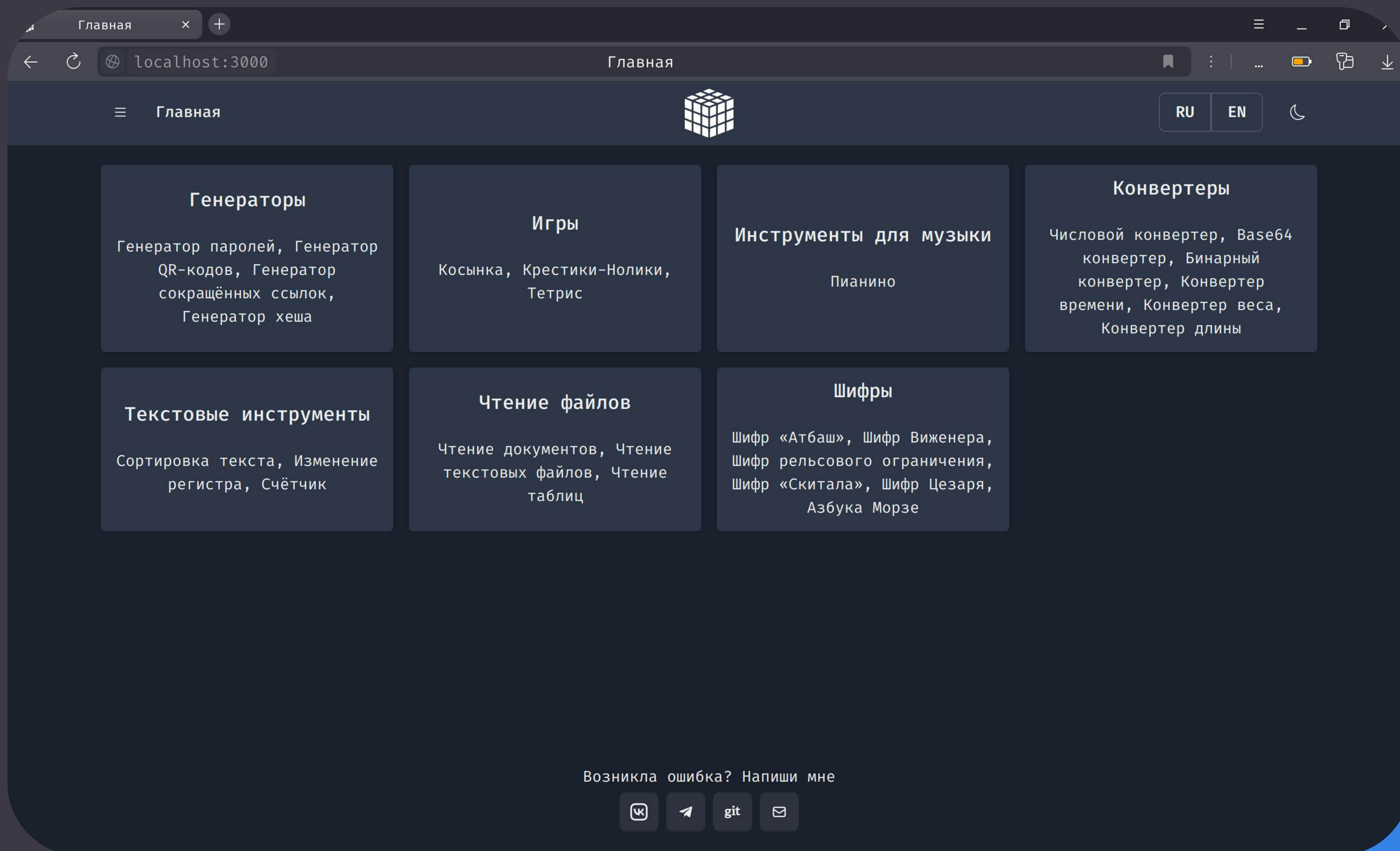
Next Steps

Next Steps

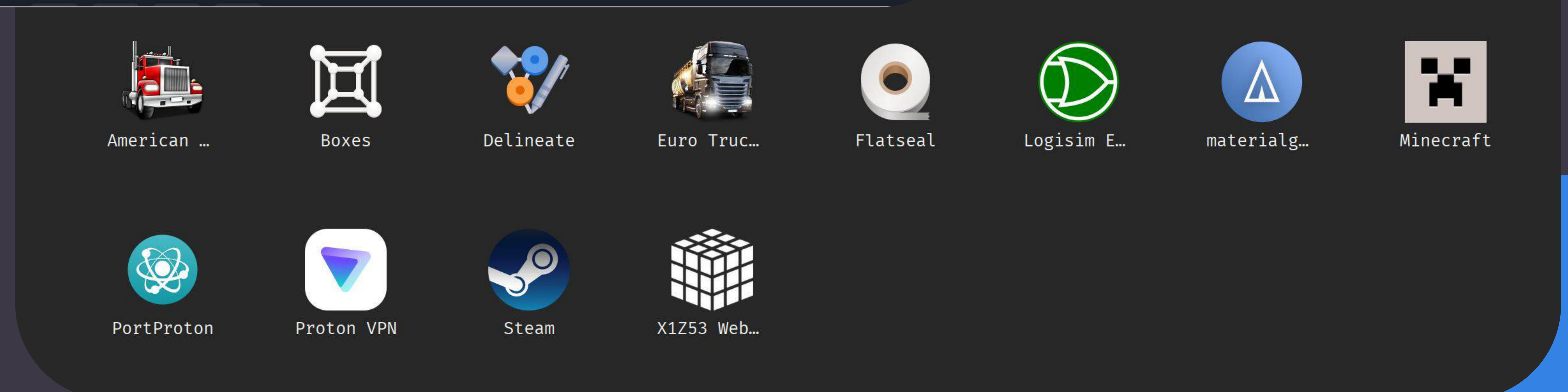
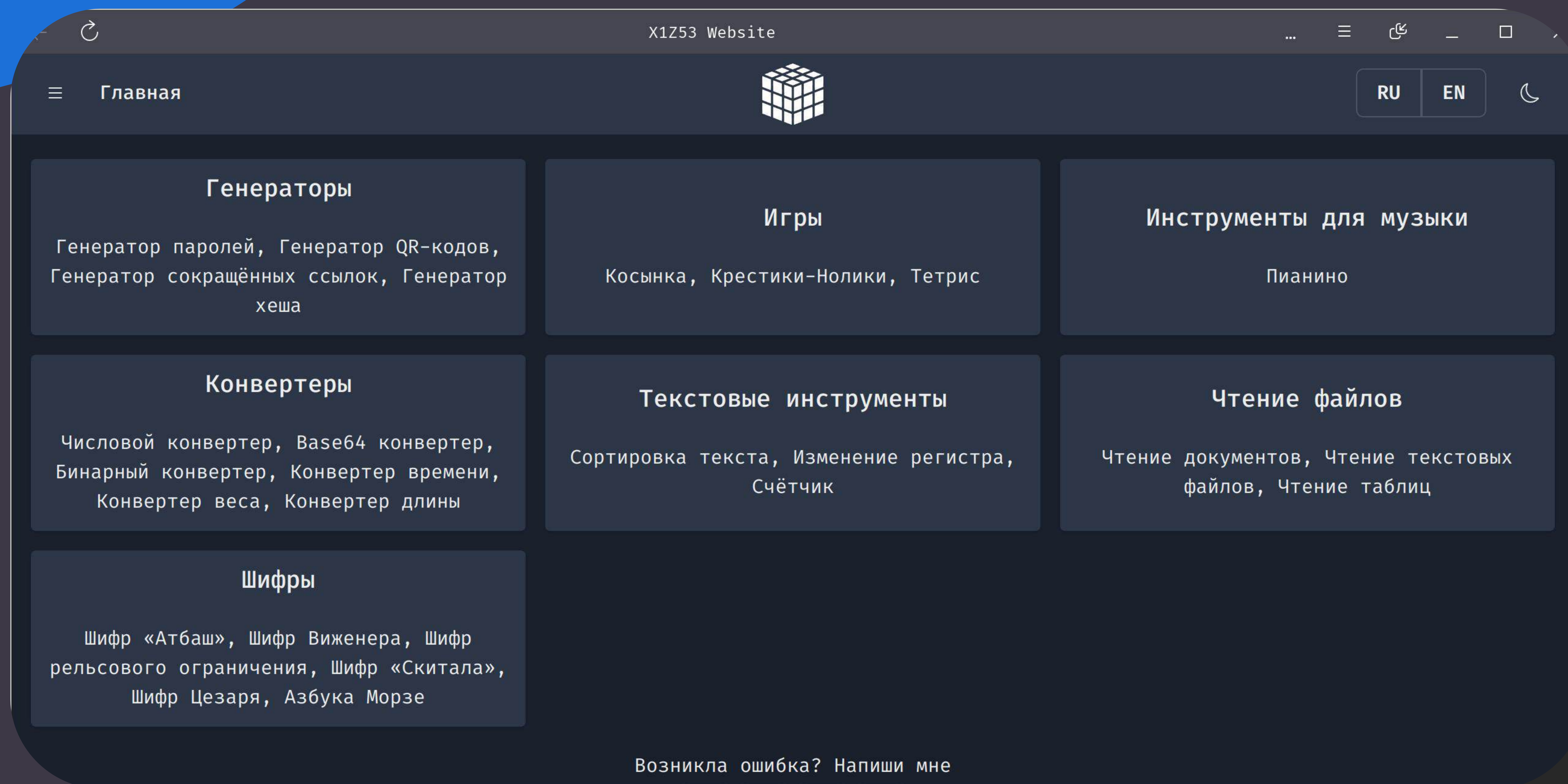
Edit this page on GitHub

Managed Next.js (Vercel)

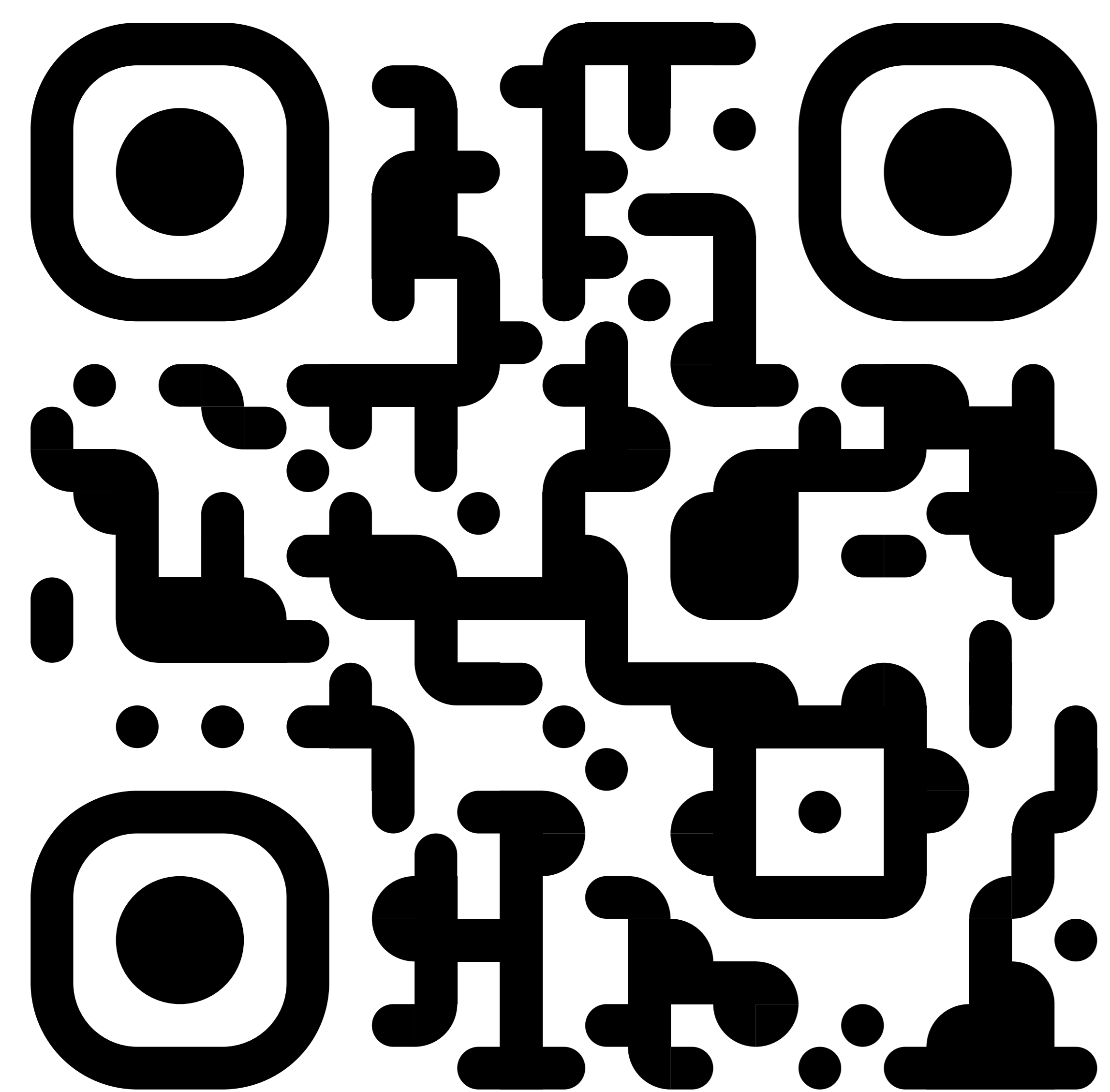
Был сайт



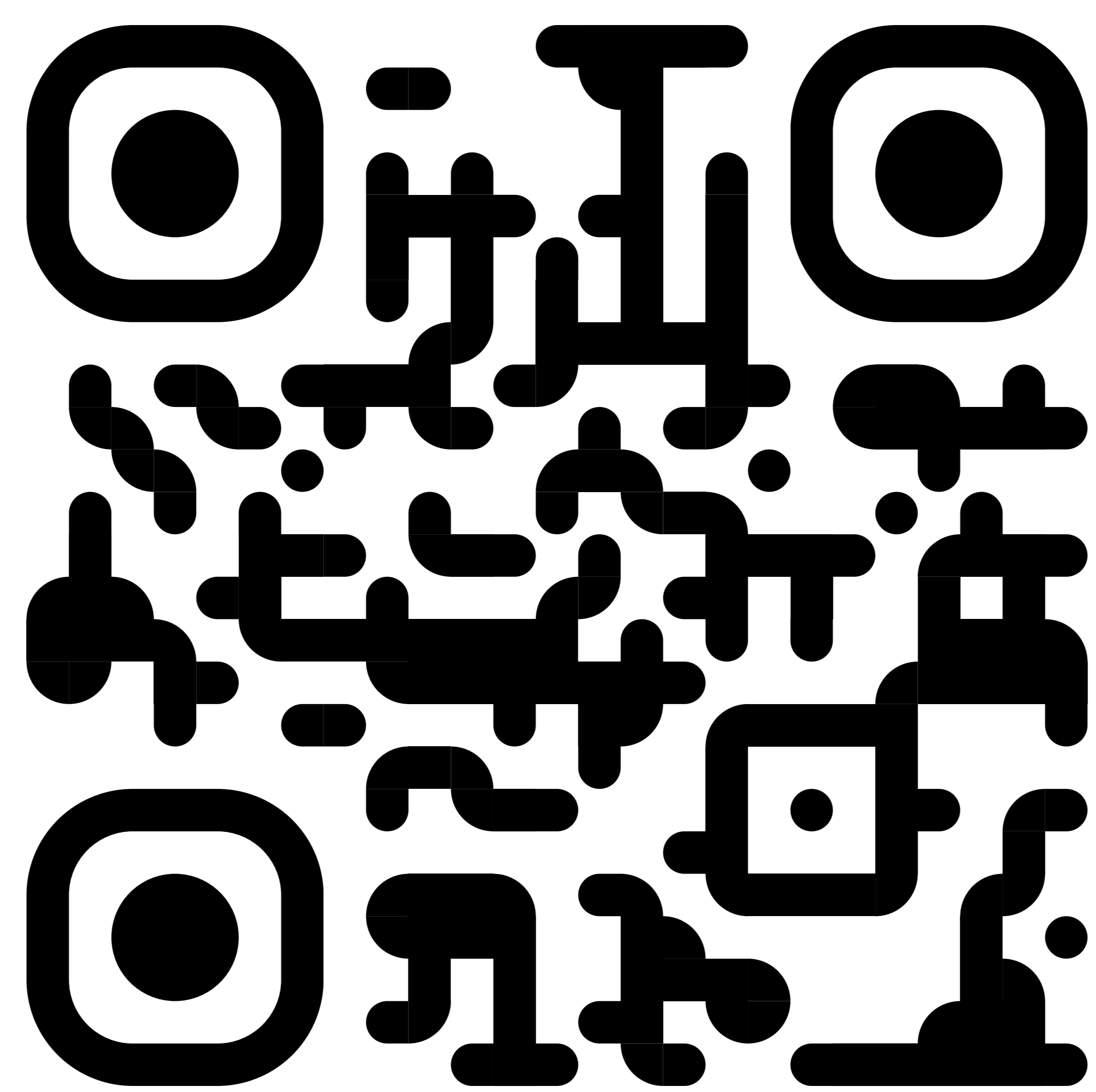
Теперь приложение



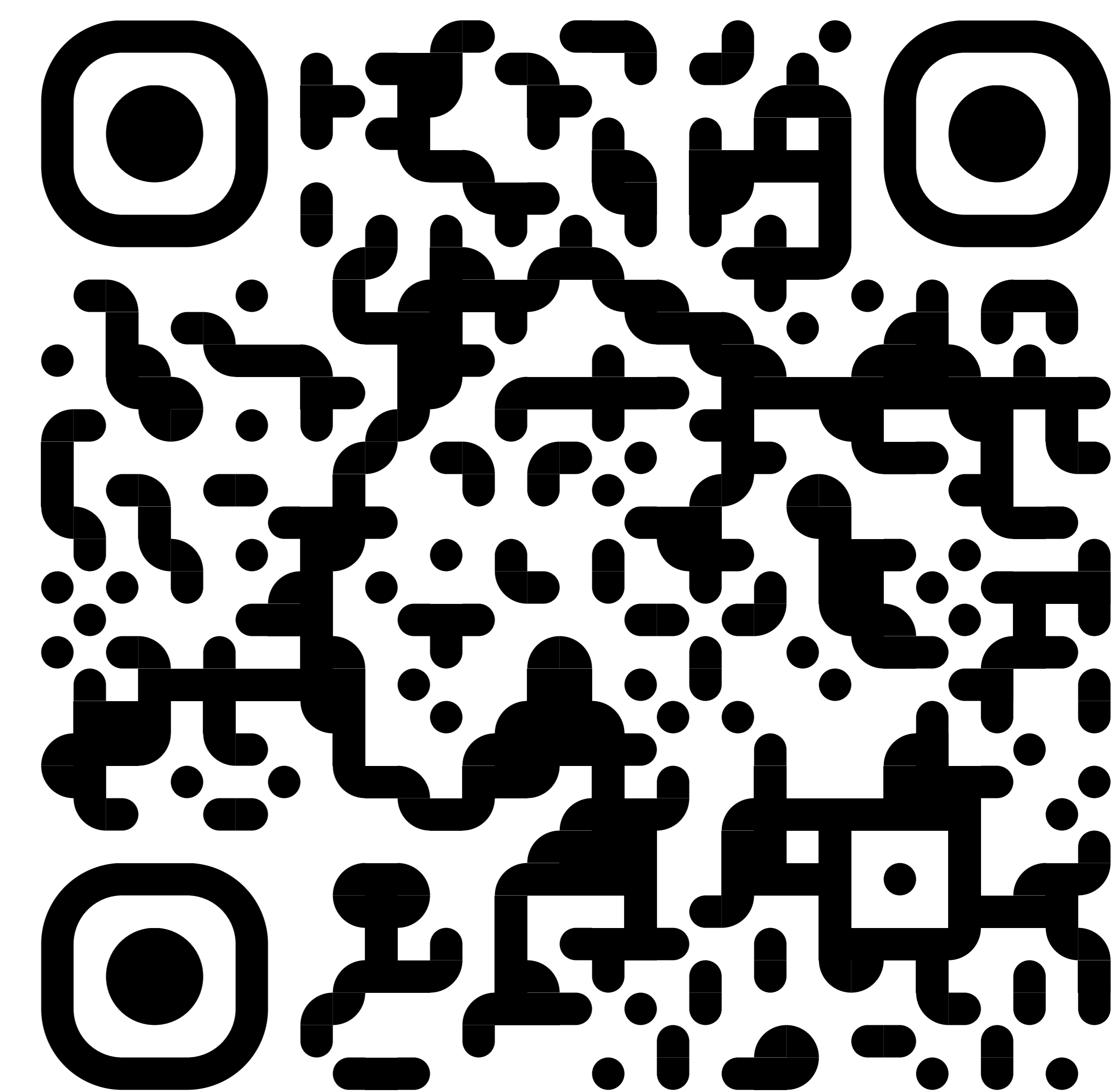
Давай к нам!



[NextJS](#)

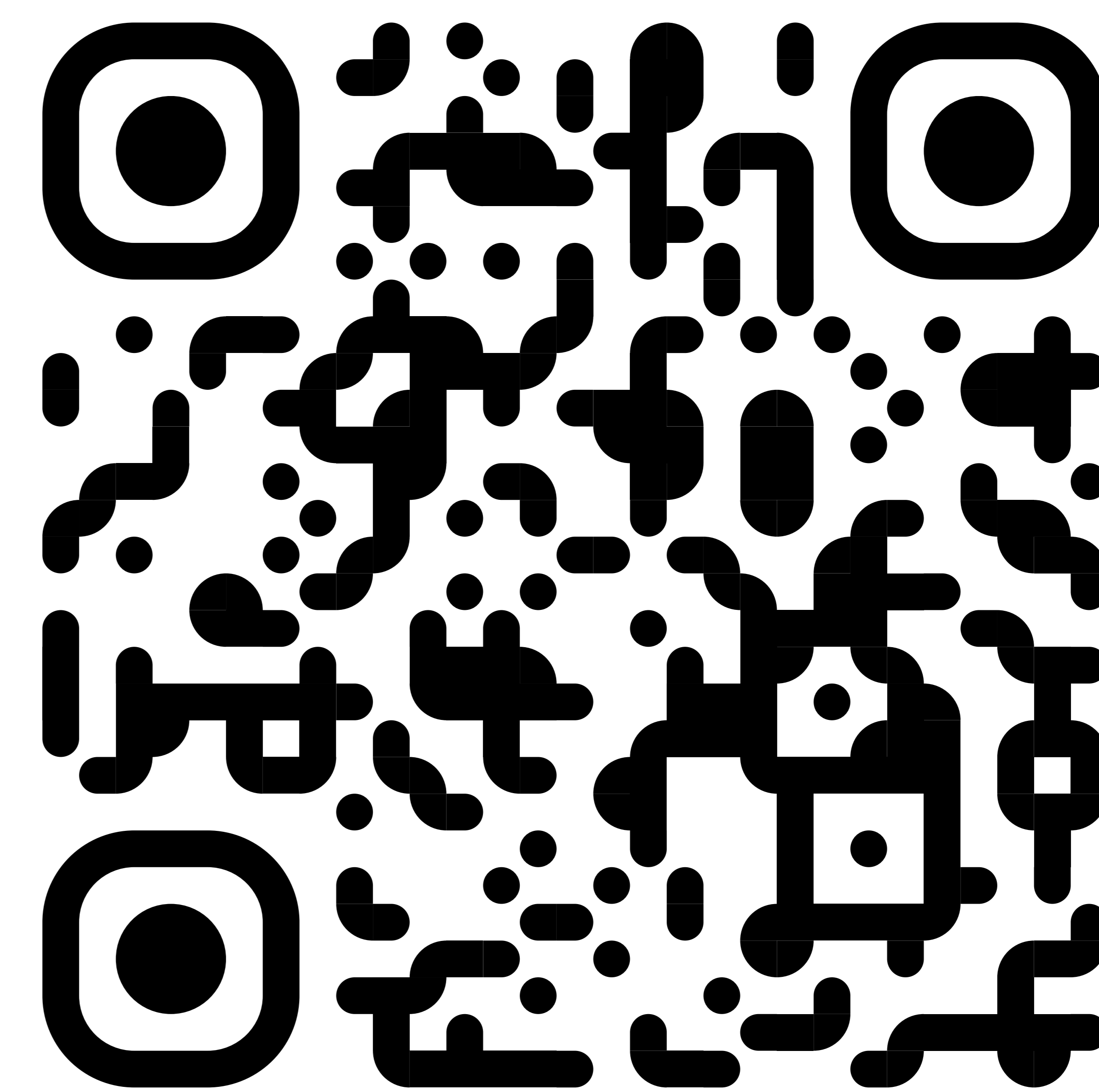


[x1z53.ru](#)



[GitVerse](#)

Спасибо за внимание!



Telegram