GenUI – Description Document

1. Overview

GenUI is an intelligent design-to-code assistant that helps developers extract visual styles from UI designs and convert them into clean, production-ready code. It bridges the gap between design and development by eliminating manual inspection, pixel measurement, and repetitive styling work.

GenUI focuses on a **task-first workflow**: extract styles → generate structured design tokens → convert to code formats such as Tailwind CSS, plain CSS, and JSX.

2. Problem Statement

Frontend developers frequently face challenges when converting UI designs into code:

- Manually measuring spacing, colors, and typography
- Rebuilding UI components from screenshots or references
- Inconsistent styles across components
- Repetitive px → rem/em conversions
- Difficulty extracting styles from interactive Figma layers due to click handlers
- Time-consuming conversions from px to rem/em and utility classes

This process is slow, error-prone, and repetitive—especially for junior developers.

3. Solution

GenUI automates the UI style extraction and conversion process.

3.1 Main Application (Web App)

A **React-based web application** that serves as the central workspace for extraction, conversion, preview, and storage.

Users can:

- Upload screenshots or images of UI designs
- Extract visual attributes such as colors, spacing, typography, and layout
- Generate a structured `design.json` file as a single source of truth
- Convert the extracted styles into Tailwind CSS, CSS, or JSX
- Extract styles from live web elements

- Paste or import JSON output from the secondary app
- Apply px → rem conversion with a configurable base
- Save and manage conversions securely per user

3.2 Secondary Application (Figma Plugin)

A lightweight **Figma plugin / Sigma extension** designed specifically to overcome Figma's interaction limitations.

- Some Figma layers block direct extraction due to click/edit handlers
- This plugin allows users to **select layers and export accurate style JSON**
- Extracted JSON can be copied or downloaded and imported into the main app

Together, these apps form a **complete design-to-code pipeline**.

4. Key Features

4.1 Web Element Style Extraction

- Extracts styles directly from rendered web elements
- Captures layout, spacing, typography, and colors
- Useful for reverse-engineering live websites and dashboards

4.2 Image-Based Style Extraction

- Supports PNG/JPG screenshots
- Uses server-side image analysis to infer styles
- Extracts:
  - Colors
  - Font size and weight (estimated)
  - Spacing and alignment
  - Layout structure
  - Component hierarchy

4.3 Figma / Sigma Design Extraction

- Uses **official Figma Plugin API**
- Allows selection of layers, frames, or components
- Extracts accurate:
  - Colors, fills, and strokes
  - Typography (font family, size, weight, line height)
  - Auto-layout rules (direction, gap, padding)
  - Border radius, shadows, and effects

    - Component and frame hierarchy
- Outputs structured JSON compatible with the main app
- Eliminates approximation issues seen in image-based extraction

4.4 Intermediate Style JSON (Design Tokens)

- Acts as a **normalized intermediate format**
- Decouples extraction from code generation
- Enables:
    - Re-conversion into multiple formats
    - Consistent styling across components
    - Future extensibility (themes, tokens, variants)

4.5 Code Conversion & Output

- Convert extracted styles into:
    - **Tailwind CSS utility classes**
    - **Plain CSS**
    - **JSX / React-compatible components**
- Toggle **px → rem conversion**
- Configurable root font-size

4.6 History & Storage

- All conversions saved per authenticated user
- Stored securely in **Zoho Catalyst Data Store**
- Users can:
    - Revisit
    - Reconvert
    - Copy or export past conversions

5. User Experience Flow

5.1 Main Application Flow

Step 1: Open Web Application
The user opens the GenUI_Style_Extractor web application and is presented with a focused interface to start the extraction process.

Step 2: Select Extraction Source
The user chooses one of the following extraction methods:

- Web element extraction
- Image upload

- Paste JSON (imported from the Figma/Sigma plugin)

Step 3: Style Extraction & Preview
Based on the selected source, GenUI extracts the UI styles and displays a structured preview, including normalized style tokens and JSON data.

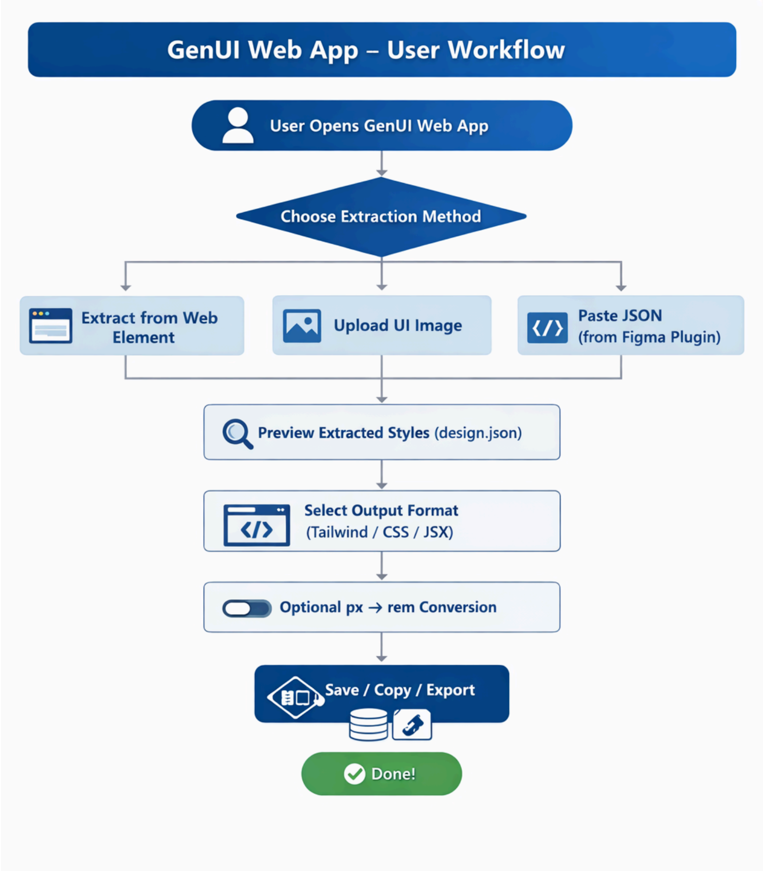
Step 4: Select Output Format
The user selects the desired output format:

- Tailwind CSS
- Plain CSS
- JSX / React-compatible code

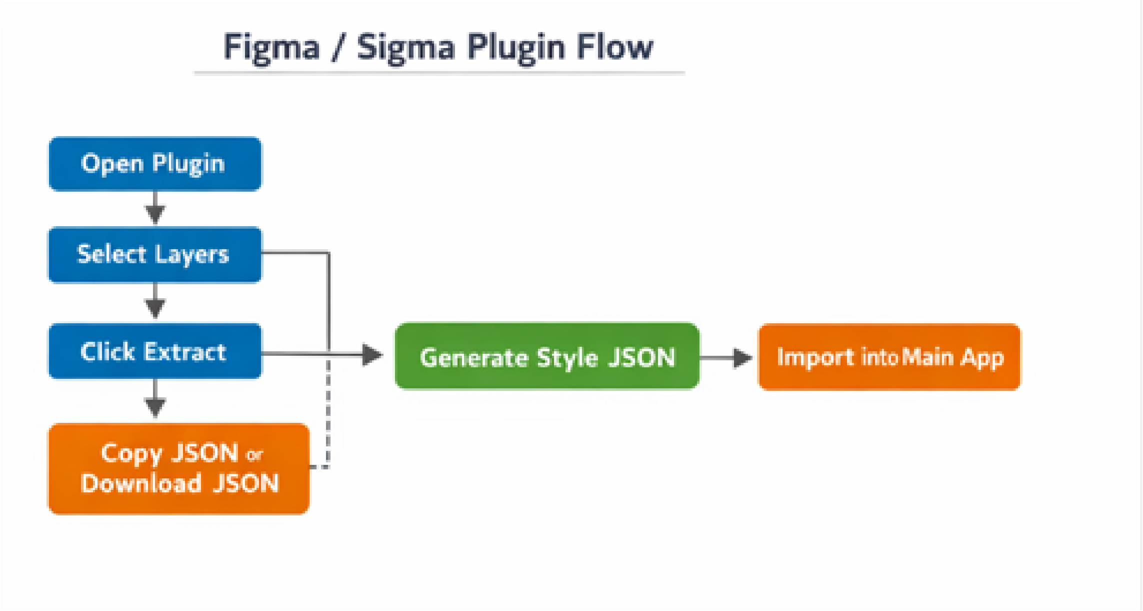
Step 5: Optional Unit Conversion
The user can enable px → rem conversion with a configurable base font size.

Step 6: Save, Copy, or Export
The final generated output can be:

- Saved to the user's history
- Copied to the clipboard
- Exported for external use


GenUI Web App – User Workflow
User Opens GenUI Web App
Choose Extraction Method
Extract from Web Element
Upload UI Image
Paste JSON (from Figma Plugin)
Preview Extracted Styles (design.json)
Select Output Format (Tailwind / CSS / JSX)
Optional px → rem Conversion
Save / Copy / Export
Done!

5.2 Secondary Application (Figma Plugin) Flow



Step 1: Open Plugin
The user opens the GenUI Figma or Sigma plugin inside the design tool.

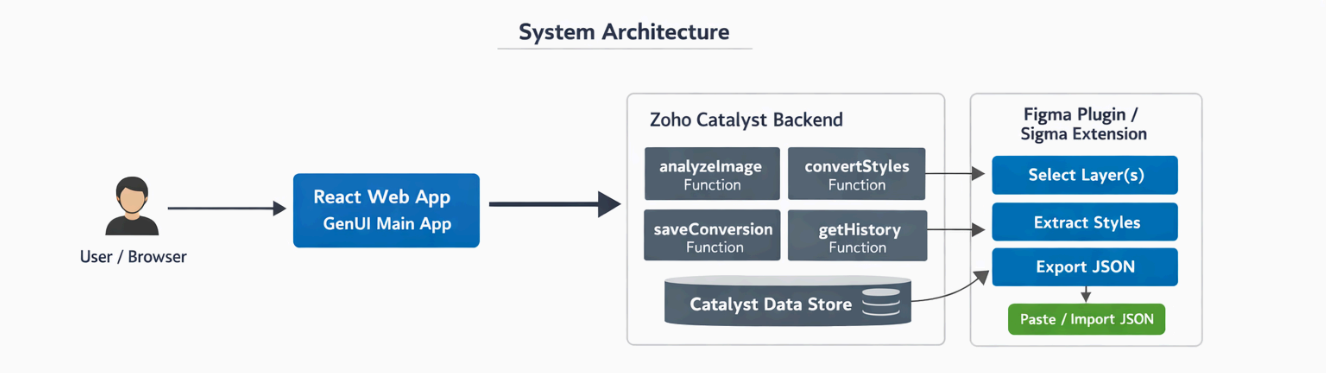
Step 2: Select Design Layers
The user selects one or more layers, frames, or components from the design canvas.

Step 3: Extract Styles
The user clicks the **Extract** action to generate a structured JSON representation of the selected styles.

Step 4: Copy or Download JSON
The extracted JSON can be copied to the clipboard or downloaded as a file.

Step 5: Import into Main Application
The user pastes or uploads the JSON into the main GenUI web application for conversion and storage.

6. Technical Implementation



6.1 Components

- **Frontend (React Web App)**
  - UI for extraction, preview, conversion, and history
- **Zoho Catalyst Backend**
  - Serverless functions for analysis, conversion, and storage
- **Figma Plugin / Sigma Extension**
  - Local extraction of accurate design properties
- **Storage**
  - Catalyst Data Store for per-user saved conversions
- **Authentication**
  - Google Sign-In via Zoho Catalyst Auth

7. Implementation Details

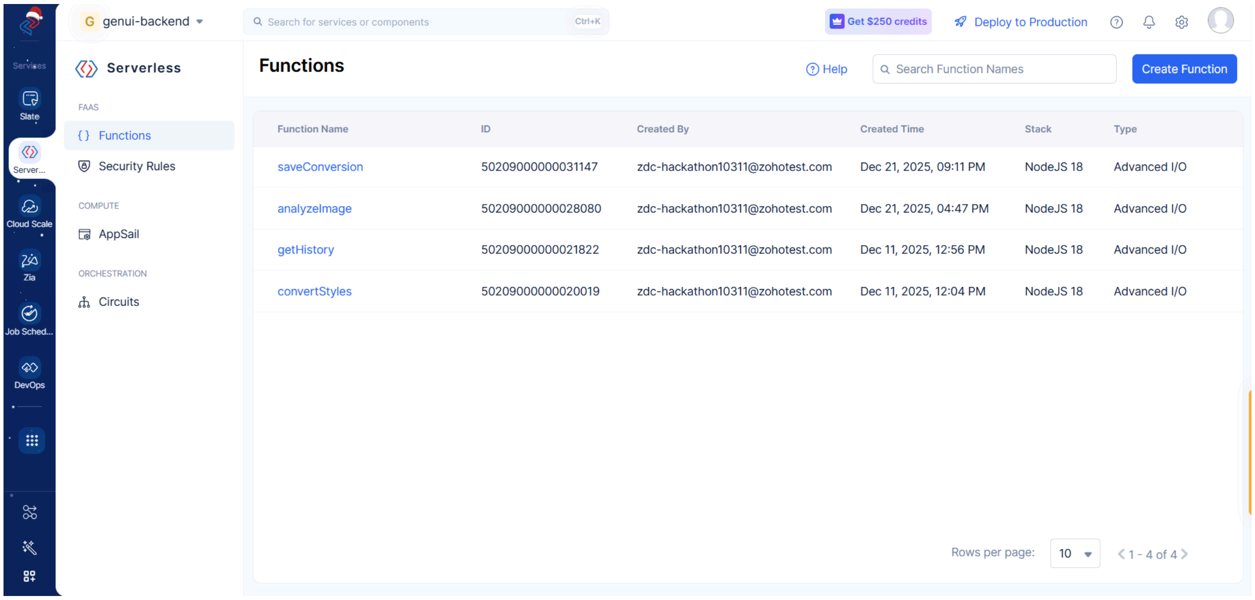
7.1 Backend Implementation (Zoho Catalyst)

7.1.1 Serverless Functions

The backend logic of **GenUI_Style_Extractor** is implemented using Zoho Catalyst **Serverless Functions (Node.js 18, Advanced I/O)**.

- `analyzeImage` – Analyzes uploaded UI images and extracts style metadata.
- `convertStyles` – Converts extracted style JSON into Tailwind CSS, CSS, or JSX.
- `saveConversion` – Stores generated code and metadata for authenticated users.
- `getHistory` – Retrieves previously saved conversions for the logged-in user.

These functions handle all processing, conversion, and persistence operations securely.



Functions

Function Name	ID	Created By	Created Time	Stack	Type
saveConversion	50209000000031147	zdc-hackathon10311@zohotest.com	Dec 21, 2025, 09:11 PM	NodeJS 18	Advanced I/O
analyzeImage	50209000000028080	zdc-hackathon10311@zohotest.com	Dec 21, 2025, 04:47 PM	NodeJS 18	Advanced I/O
getHistory	50209000000021822	zdc-hackathon10311@zohotest.com	Dec 11, 2025, 12:56 PM	NodeJS 18	Advanced I/O
convertStyles	50209000000020019	zdc-hackathon10311@zohotest.com	Dec 11, 2025, 12:04 PM	NodeJS 18	Advanced I/O

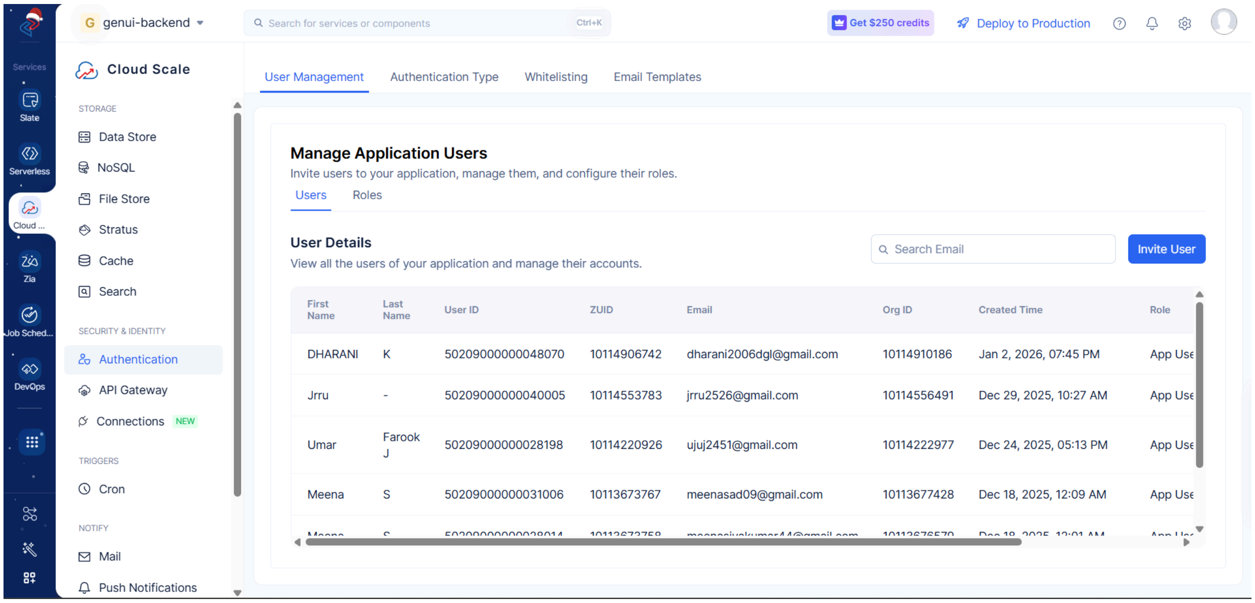
Rows per page: 10 ‹ 1 - 4 of 4 ›

7.1.2 Authentication & User Management

User authentication is managed using **Zoho Catalyst Authentication** with Google Sign-In integration.

- Users authenticate securely using their Google accounts.
- Each user is assigned a unique user ID.
- All saved conversions and history are linked to authenticated users.
- Role-based access ensures secure and isolated data per user.

This enables personalized usage and secure access control across the application.



Manage Application Users

Invite users to your application, manage them, and configure their roles.

User Details

View all the users of your application and manage their accounts.

First Name	Last Name	User ID	ZUID	Email	Org ID	Created Time	Role
DHARANI	K	50209000000048070	10114906742	dharani2006dgl@gmail.com	10114910186	Jan 2, 2026, 07:45 PM	App Use
Jrru	-	50209000000040005	10114553783	jrru2526@gmail.com	10114556491	Dec 29, 2025, 10:27 AM	App Use
Umar	Farook J	50209000000028198	10114220926	ujuj2451@gmail.com	10114222977	Dec 24, 2025, 05:13 PM	App Use
Meena	S	50209000000031006	10113673767	meenasad09@gmail.com	10113677428	Dec 18, 2025, 12:09 AM	App Use
Meena	S	50209000000028014	10113673758	meenasivakumar44@gmail.com	10113676579	Dec 18, 2025, 12:01 AM	App Use

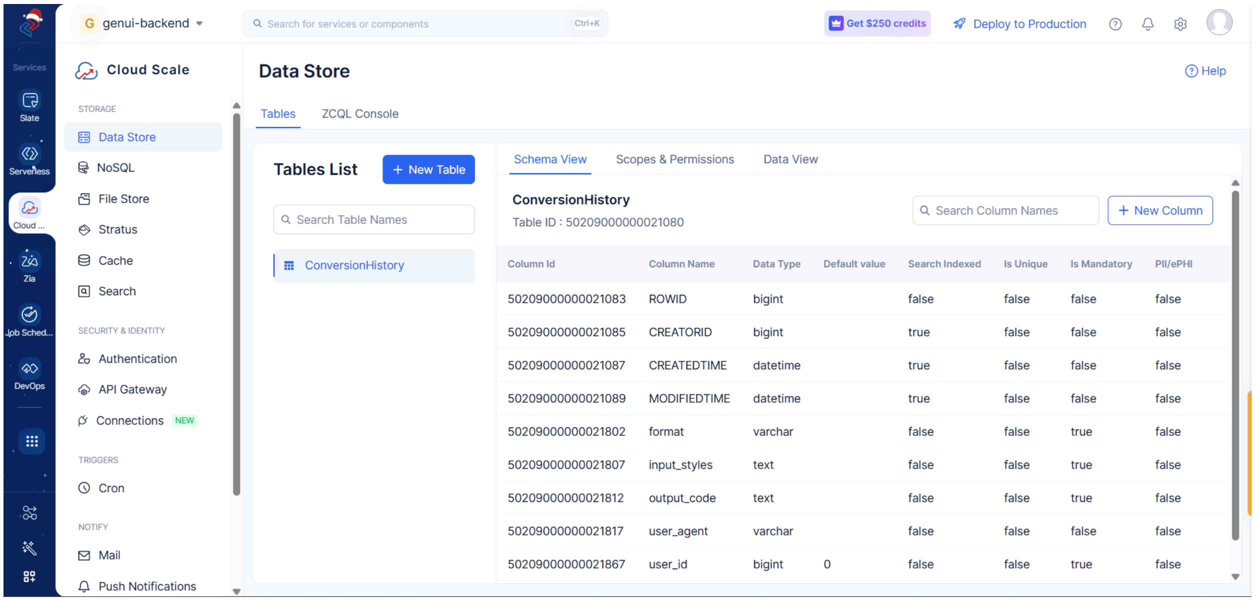
7.1.3 Data Store (Database Design)

GenUI uses **Zoho Catalyst Data Store** to persist user conversions.

ConversionHistory Table stores:

- User ID (owner of the conversion)
- Input styles (JSON)
- Output code (CSS / JSX / Tailwind)
- Conversion format
- Created and modified timestamps

This structure allows efficient retrieval and management of conversion history per user.



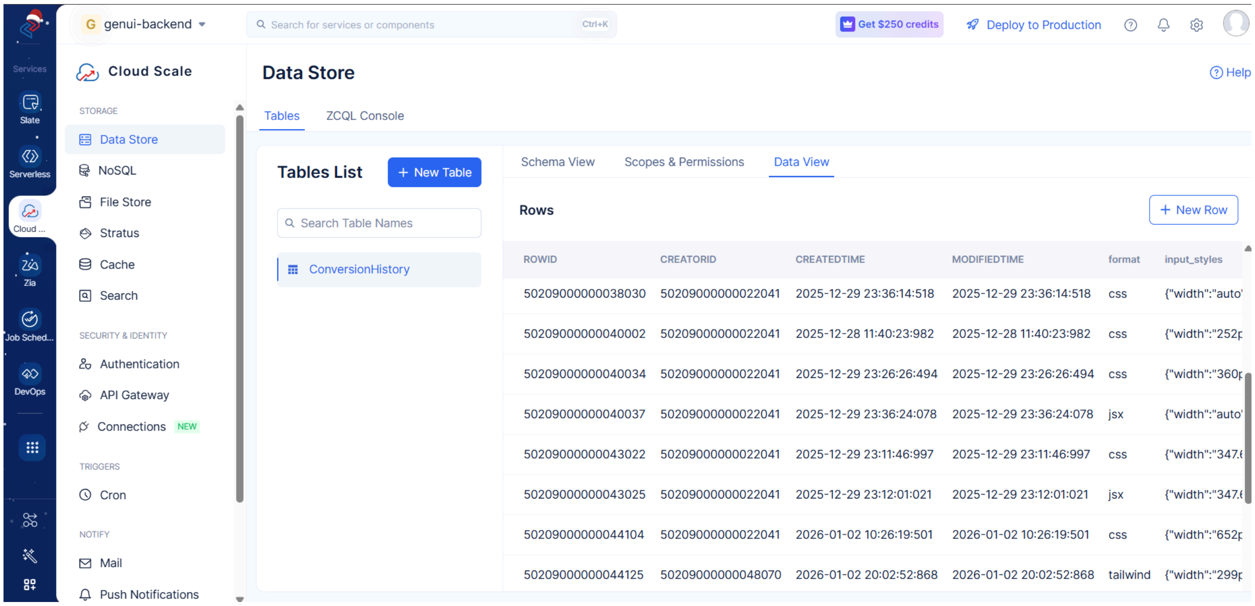
ConversionHistory
Table ID : 50209000000021080

Column Id	Column Name	Data Type	Default value	Search Indexed	Is Unique	Is Mandatory	PII/ePHI
50209000000021083	ROWID	bigint		false	false	false	false
50209000000021085	CREATORID	bigint		true	false	false	false
50209000000021087	CREATEDTIME	datetime		true	false	false	false
50209000000021089	MODIFIEDTIME	datetime		true	false	false	false
50209000000021802	format	varchar		false	false	true	false
50209000000021807	input_styles	text		false	false	true	false
50209000000021812	output_code	text		false	false	true	false
50209000000021817	user_agent	varchar		false	false	false	false
50209000000021867	user_id	bigint	0	false	false	true	false

7.1.4 Stored Conversion Records

The Data View confirms successful storage of conversion records.

- Each row represents a completed conversion
- Supports multiple formats (CSS, JSX, Tailwind)
- Enables history, reuse, and reconversion features in the UI
- This validates the end-to-end flow from extraction to persistent storage.



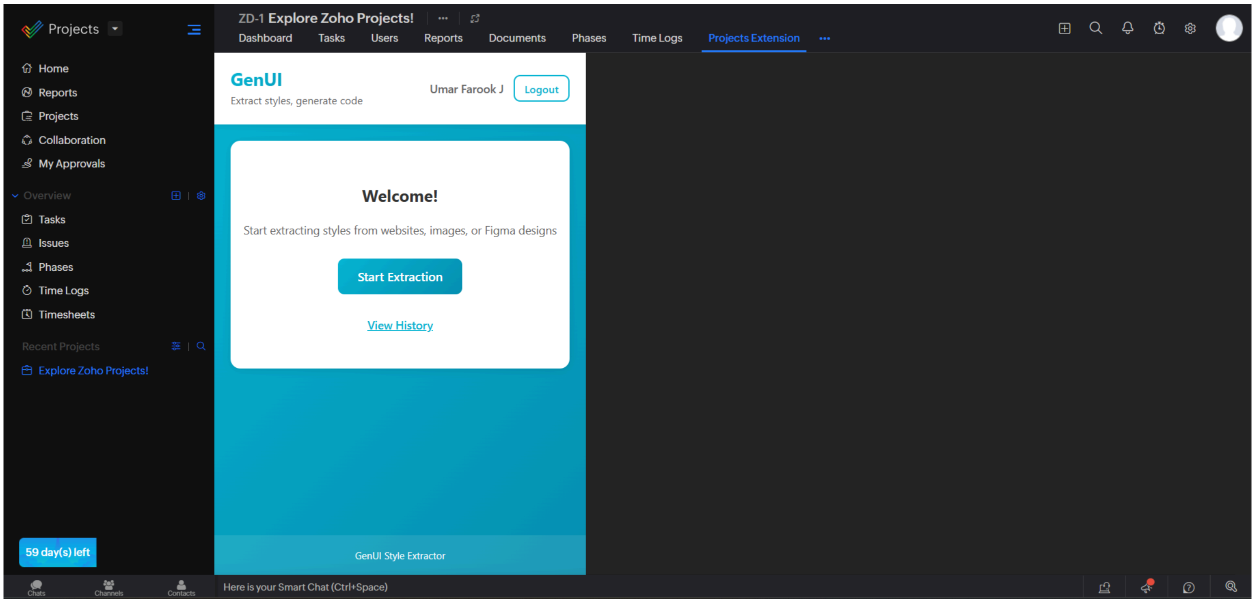
ROWID	CREATORID	CREATEDTIME	MODIFIEDTIME	format	input_styles
50209000000038030	50209000000022041	2025-12-29 23:36:14:518	2025-12-29 23:36:14:518	css	{"width":"auto'
50209000000040002	50209000000022041	2025-12-28 11:40:23:982	2025-12-28 11:40:23:982	css	{"width":"252p
50209000000040034	50209000000022041	2025-12-29 23:26:26:494	2025-12-29 23:26:26:494	css	{"width":"360p
50209000000040037	50209000000022041	2025-12-29 23:36:24:078	2025-12-29 23:36:24:078	jsx	{"width":"auto'
50209000000043022	50209000000022041	2025-12-29 23:11:46:997	2025-12-29 23:11:46:997	css	{"width":"347.6
50209000000043025	50209000000022041	2025-12-29 23:12:01:021	2025-12-29 23:12:01:021	jsx	{"width":"347.6
50209000000044104	50209000000022041	2026-01-02 10:26:19:501	2026-01-02 10:26:19:501	css	{"width":"652p
50209000000044125	50209000000048070	2026-01-02 20:02:52:868	2026-01-02 20:02:52:868	tailwind	{"width":"299p

7.2 Frontend

- Built with **React (TypeScript) + Vite**
- Routing via **HashRouter**
- Modular components for extraction and preview

7.3 Extensions

- Sigma extension packaged for marketplace submission



Welcome!

Start extracting styles from websites, images, or Figma designs

8. Main Application Screens

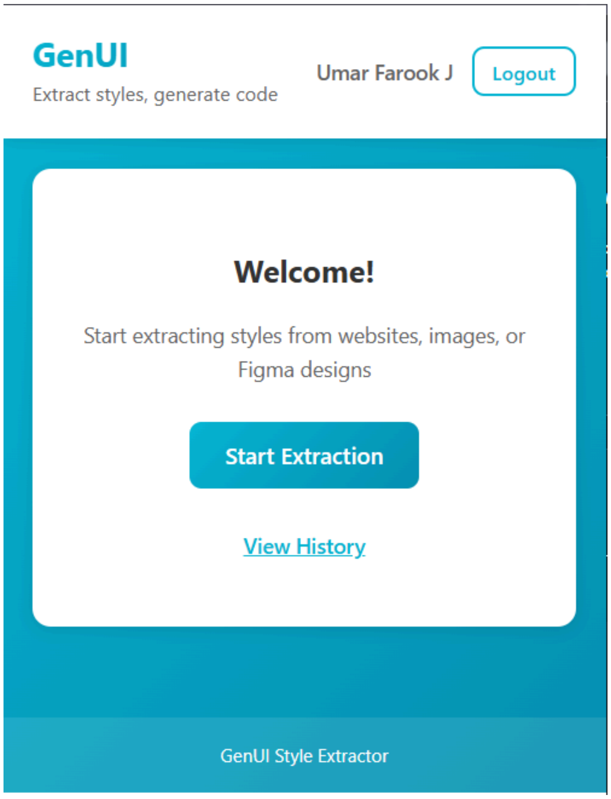
The following screens represent the core workflow of the **GenUI_Style_Extractor main web application**:

Home Screen:

Entry point after login with options to start a new extraction or view past conversions.

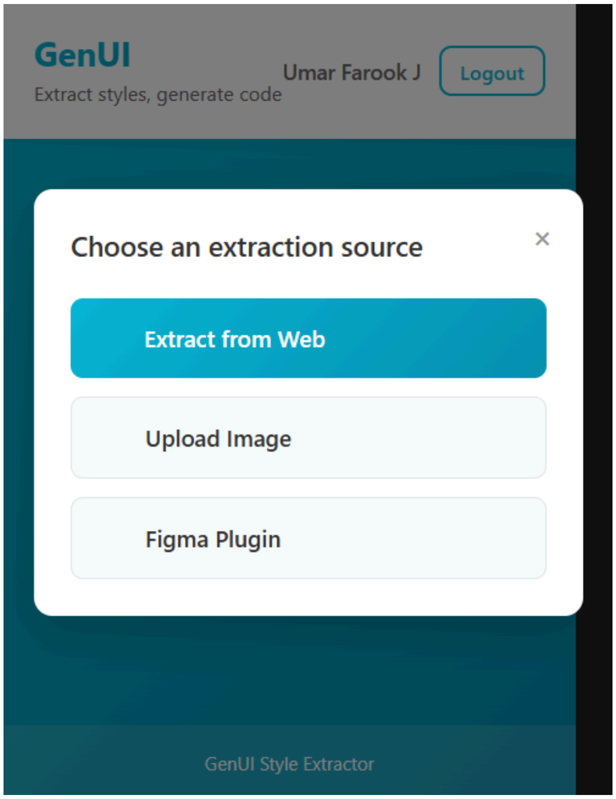
Extraction Source Selection:

Modal to choose extraction method – Web Element, Image Upload, or Figma Plugin JSON.



Welcome!

Start extracting styles from websites, images, or Figma designs



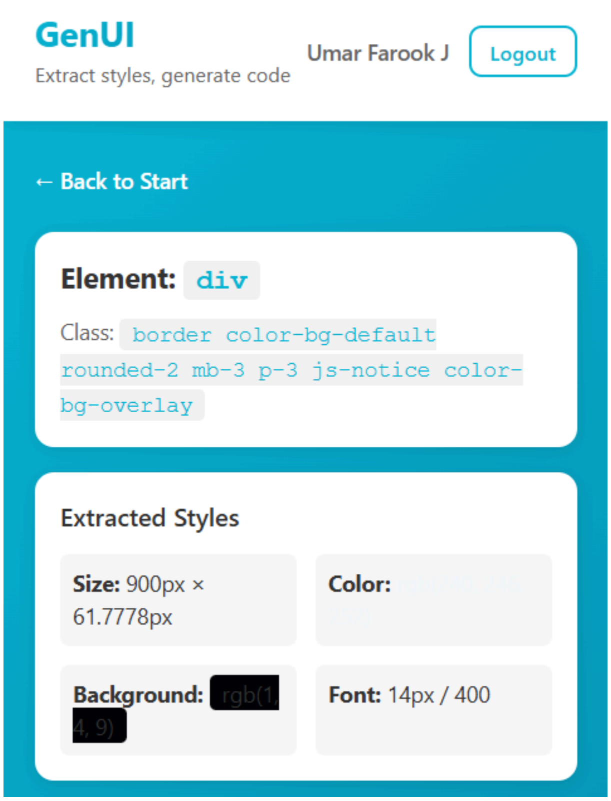
Choose an extraction source

Web Element Extraction:

Displays selected HTML element details and extracted styles such as size, color, background, and typography.

Image Upload:

Allows users to upload UI screenshots for image-based style extraction.



GenUI

Extract styles, generate code

Umar Farook J

Element: `div`

Class: `border color-bg-default rounded-2 mb-3 p-3 js-notice color-bg-overlay`

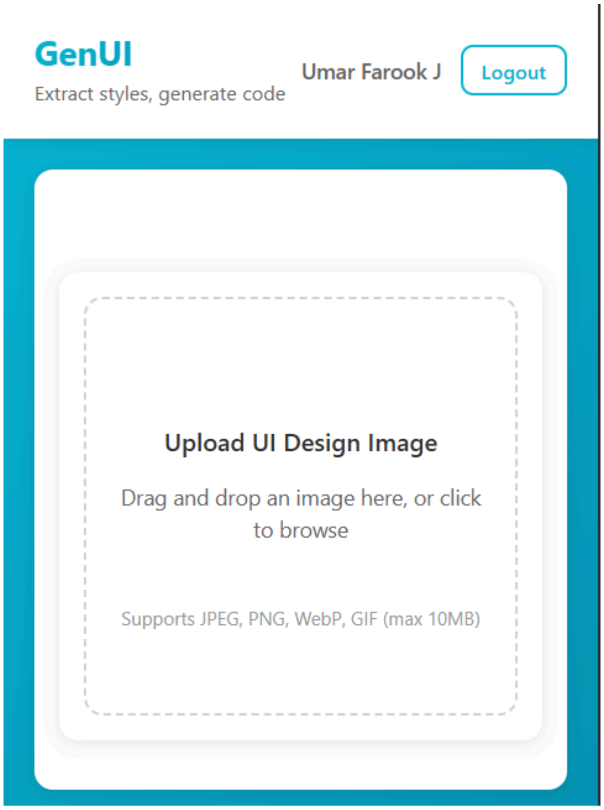
Extracted Styles

Size: 900px × 61.7778px

Color:

Background: rgb(1, 4, 9)

Font: 14px / 400



GenUI

Extract styles, generate code

Umar Farook J

Upload UI Design Image

Drag and drop an image here, or click to browse

Supports JPEG, PNG, WebP, GIF (max 10MB)

Design JSON View:

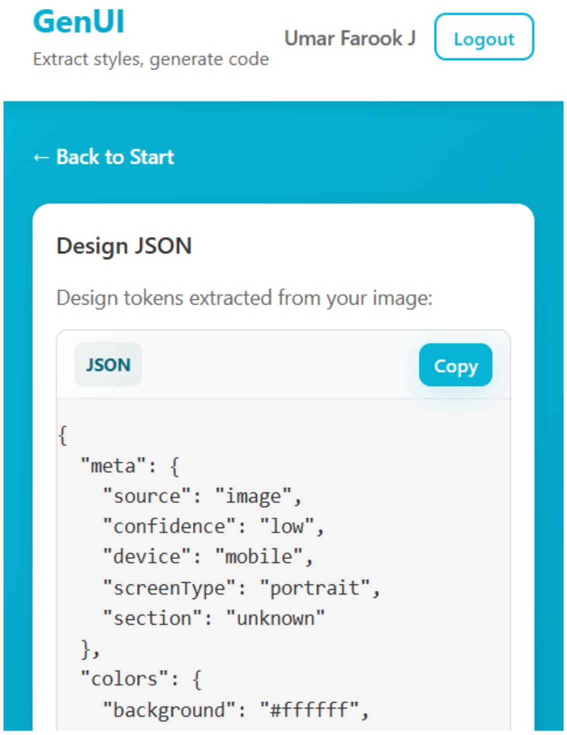
Shows normalized design tokens generated from extraction, with copy functionality.

Conversion Options:

Users select output format (Tailwind CSS, CSS, JSX) and optionally enable px → rem conversion.

Generated Code Output:

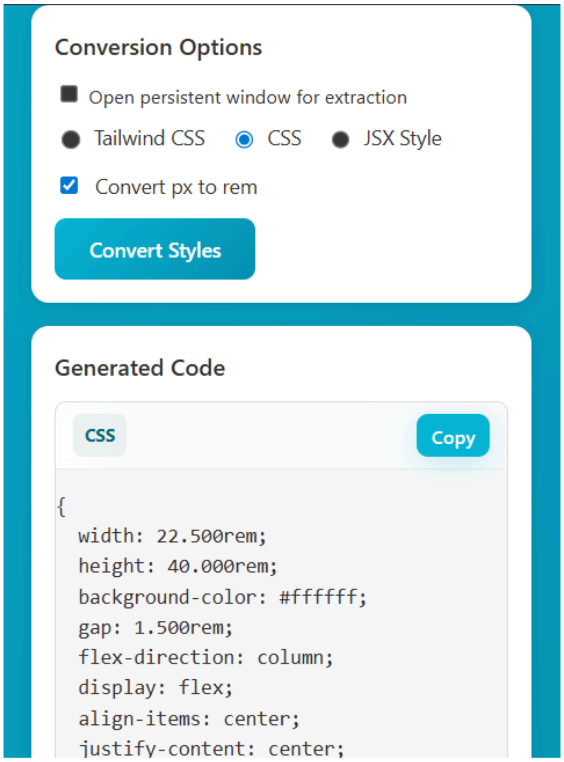
Displays production-ready code with copy support.



Design JSON

Design tokens extracted from your image:

```
{
  "meta": {
    "source": "image",
    "confidence": "low",
    "device": "mobile",
    "screenType": "portrait",
    "section": "unknown"
  },
  "colors": {
    "background": "#ffffff",
```



Conversion Options

■ Open persistent window for extraction

● Tailwind CSS ◉ CSS ● JSX Style

☑ Convert px to rem

Generated Code

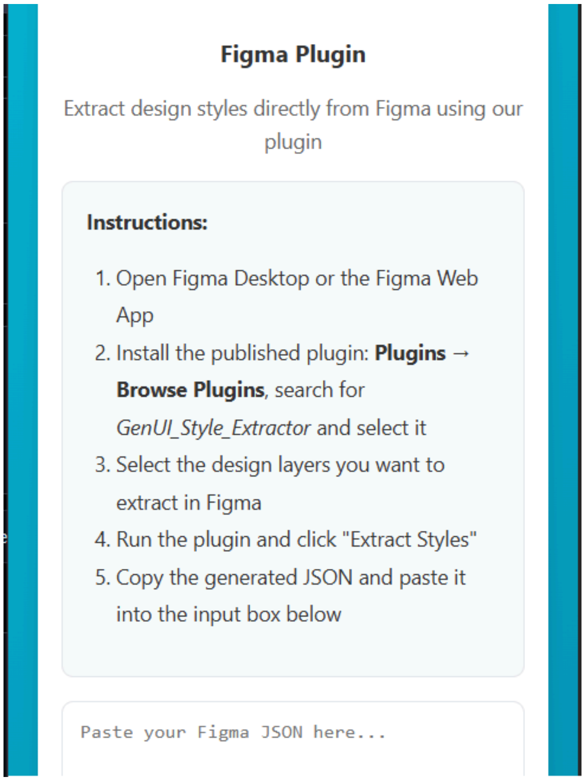
```
{
  width: 22.500rem;
  height: 40.000rem;
  background-color: #ffffff;
  gap: 1.500rem;
  flex-direction: column;
  display: flex;
  align-items: center;
  justify-content: center;
```

Conversion History:

Lists previously saved conversions per user for reuse and reconversion.

Figma Plugin Integration:

Provides instructions and input area to paste JSON generated from the Figma plugin.



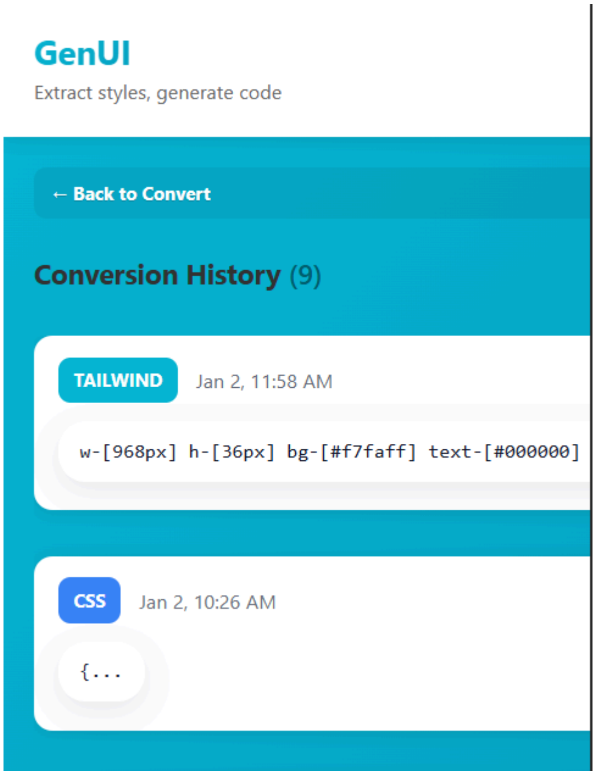
Figma Plugin

Extract design styles directly from Figma using our plugin

Instructions:

1. Open Figma Desktop or the Figma Web App
2. Install the published plugin: **Plugins → Browse Plugins**, search for *GenUI_Style_Extractor* and select it
3. Select the design layers you want to extract in Figma
4. Run the plugin and click "Extract Styles"
5. Copy the generated JSON and paste it into the input box below

Paste your Figma JSON here...



GenUI

Extract styles, generate code

Conversion History (9)

Jan 2, 11:58 AM

```
w-[968px] h-[36px] bg-[#f7faff] text-[#000000]
```

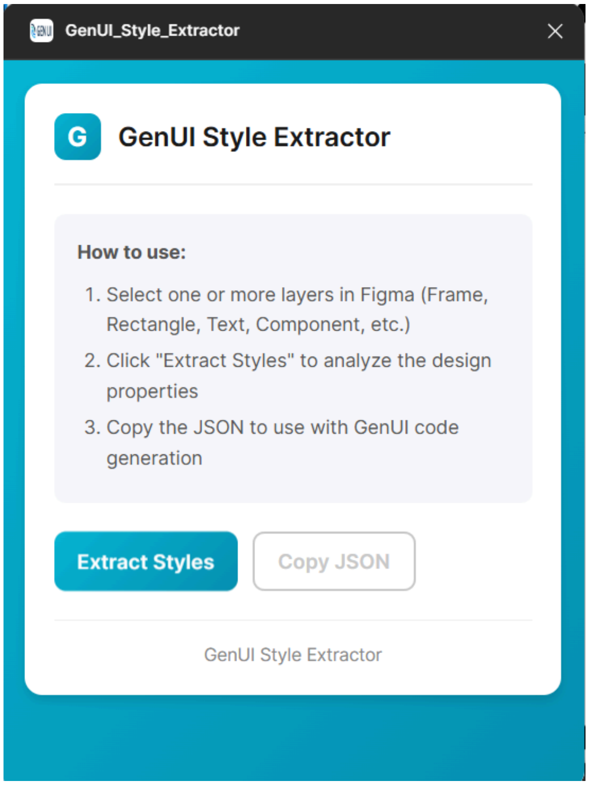
Jan 2, 10:26 AM

```
{...
```

Figma Plugin Screens

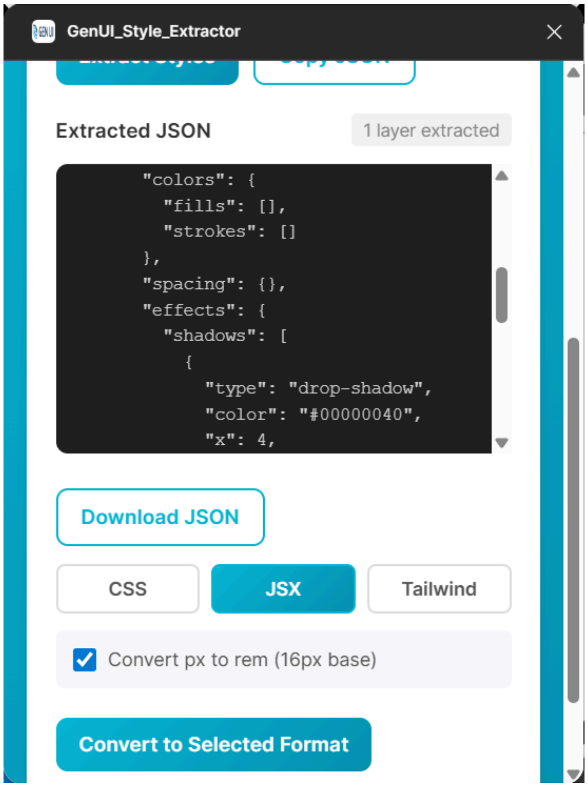
Home Screen
Allows users to select Figma layers and extract accurate design styles using a simple guided interface.

Extracted JSON & Conversion Options
Displays the extracted style JSON with options to download, copy, or convert it into CSS, JSX, or Tailwind formats.



How to use:

1. Select one or more layers in Figma (Frame, Rectangle, Text, Component, etc.)
2. Click "Extract Styles" to analyze the design properties
3. Copy the JSON to use with GenUI code generation



```
    "colors": {
      "fills": [],
      "strokes": []
    },
    "spacing": {},
    "effects": {
      "shadows": [
        {
          "type": "drop-shadow",
          "color": "#00000040",
          "x": 4,
```

☑ Convert px to rem (16px base)

9. Authentication & Security

- Authentication handled via **Zoho Catalyst Auth**
- Google OAuth used for secure sign-in
- Catalyst issues secure tokens for API calls
- All write operations validate user identity
- Data stored strictly **per user**, ensuring privacy and isolation

10. Deployment & Distribution

- Frontend hosted as a **static web client**
- Backend deployed in **Zoho Catalyst serverless environment**

- Figma plugin distributed via manifest import
- Sigma extension submitted and **currently under review**

11. Conclusion

GenUI_Style_Extractor significantly simplifies and accelerates UI development by transforming visual designs into **structured, reusable, and production-ready code**.

By combining:

- Multi-source extraction
- A normalized intermediate JSON format
- Powerful conversion options
- Secure cloud storage

GenUI provides a scalable and future-ready foundation for modern **design-to-code automation**, reducing development time, improving consistency, and empowering developers to focus on building features—not recreating styles.