**Beginning USER JOURNEY**

**SEO USER WORKFLOWS**

**User Journey**

Introduction to **GEN-AIS’s AI-driven SEO automation**. WITH A WELCOME STATEMENT. Two Buttons (1) NEW USER and (2) Returning USER

**For a NEW USER enters the GEN-AIS portal and enters their URL for a Free Assessment using the URL SEO Analysis Tool** – Users input their domain for AI-powered assessment.

**\*\*\*Concurs to the review. If agreed, the USER is directed to the Registration entry point. Else END!\*\*\***

**SEO tool generates the results for display ONLY.**

**TO Continue NEW USER MUST Register.**

**Directs the new user to a Registration Feature**

**RETURNING User Login Flow**

* ✅ **Step 1: User Clicks on SEO Tool Link** – Redirects to **Login Portal**. ✅ **Step 2: Login** – Users select **Existing Account**. ✅ **Step 3: Authentication Processing** – Validate credentials securely against **GEN-AIS's User Database**. ✅ **Step 4: Redirect to Dashboard** – Once logged in, users access **SEO tracking, audit reports, and automation settings**.

✅ **Post-Login Redirection** – Successful authentication redirects users to the **SEO AI Service dashboard**. ✅ **Navigation Panel** – Provides easy access to:

**User Interaction Flow**

[🔘 **Run New SEO Scan**] [🔘 **View Detailed Report**] [🔘 **Optimize Site Now**]

After the SCAN

Users receive an **SEO audit report**, showing optimization recommendations.

If metadata updates are needed, users select either:

1. **Auto AI SEO Update** → AI automatically adjusts metadata based on ranking intelligence.

**SEO Audit Reports** – Displays real-time metadata insights, ranking recommendations, and optimization status. **Site Analysis Reports & Keyword Recommendations**

**AI Automation Settings** – Users execute **structured optimizations and keyword refinements**. **SEO Tool Execution & Automation Controls**

**(B) Proceed with AI SEO Op** → AI guides users through manual optimizations.

**SUPPLEMENT INTERACTION**

* **Account & Security Controls** – Manage profile settings, logout options, and security preferences. ✅ **Tier-Based Access Flow** – Ensures Basic, Advanced, and Enterprise users **see relevant features** for their subscription level. **Account Settings & Subscription Management Subscription Management** – Allows upgrades, renewals, and feature expansion. **Logout & Security Features** – Ensures session safety and account management.
* **Technical Implementation**
* 1️⃣ **Modify Landing Page Click Behavior** – Ensure the SEO tool section **triggers the login page first**. 2️⃣ **Develop Secure Login System** – Store user credentials in a **GEN-AIS-specific database**. 3️⃣ **Create User Dashboard Access Controls** – Assign privileges based on **subscription tiers** (Basic, Advanced, Enterprise). 4️⃣ **Session Handling & Security** – Maintain **secure authentication** and prevent unauthorized access.

**Designing GEN-AIS's User Dashboard & Access Controls**

**Returning User Dashboard Layout**

**Welcome Screen** – Displays user name, subscription tier, and latest SEO insights. ✅ **URL SEO Analysis Tool** – Validate **website URL** (One to One relationship) for single user for ✅ **SEO Audit Reports** – Displays **real-time ranking updates, metadata issues, and optimization suggestions**. ✅ **AI Automation Controls** – Users execute **metadata updates, keyword refinements, and structured optimizations**.

**Backend Flow for Dashboard Access**

1️⃣ **Session Validation** – Users are redirected to the dashboard **only after successful login**. 2️⃣ **Tier-Based Feature Activation** – Determines **dashboard capabilities** based on subscription level. 3️⃣ **SEO Tool Integration** – Allows users to run **site analysis and optimization recommendations**. 4️⃣ **User Account Settings** – Users **manage profile info, security preferences, and subscription details**. 5️⃣ **Data Storage & Retrieval** – Saves user interactions, SEO results, and settings in the **database**.

**Finalizing Dashboard Navigation for GEN-AIS Users**

**Once database, login system, and session handling are prepared**, the **dashboard navigation** will ensure a **seamless, user-friendly flow** before integration.

**Dashboard Navigation Flow**

✅ **Landing Page Entry** – Users arrive at https://www.gen-ais.com/seo-ai-service.

✅ **Login Redirect** – Clicking the **SEO Tool section** directs users to the **Login Page**. ✅ **Authentication Process** – Users either log in or register for an account.

✅ **Access Control Based on Subscription Tier** – Dashboard features **vary based on Basic, Advanced, Enterprise plans**.

✅ **SEO Audit & Optimization Panel** – Users view **SEO reports, metadata insights, and automation controls**.

**GEN-AIS User Dashboard Mockup**

**[Header Section]** 💡 Welcome, **[User Name]**! You have **[Subscription Tier] Access**.

📈 **Current Site SEO Status** 🔹 Last Scan: **[Timestamp]** 🔹 URL Analysis: **[User Website]**

**[Navigation Panel]** 📝 **Site Compatibility Check** → Enter URL to analyze SEO readiness 📊 **SEO Audit Reports** → View real-time recommendations for optimization

⚙️ **AI Automation Controls** → Apply suggested **metadata updates & keyword refinements** 💳 **Subscription Plan Management** → Upgrade/downgrade AI automation levels 🔒 **Account & Security Settings** → Manage **login credentials & session settings**

**Backend Implementation (Python & Flask)**

Suggestion on how to handle **session management** for users:

from flask import Flask, request, session, redirect, url\_for

import bcrypt

import sqlite3

app = Flask(\_\_name\_\_)

app.secret\_key = 'secure\_genais\_session'

@app.route('/login', methods=['POST'])

def login():

email = request.form['email']

password = request.form['password']

conn = sqlite3.connect('genais\_users.db')

cursor = conn.cursor()

cursor.execute("SELECT password\_hash, subscription\_tier FROM users WHERE email=?", (email,))

result = cursor.fetchone()

if result and bcrypt.checkpw(password.encode('utf-8'), result[0]):

session['user\_email'] = email

session['subscription\_tier'] = result[1]

return redirect(url\_for('dashboard'))

else:

return "Invalid credentials. Please try again."

@app.route('/dashboard')

def dashboard():

if 'user\_email' in session:

return f"Welcome, {session['user\_email']}! You have {session['subscription\_tier']} access."

return redirect(url\_for('login'))

if \_\_name\_\_ == "\_\_main\_\_":

app.run(debug=True)