

Survivor's Dashboard – Build & Deployment Documentation

This document summarizes all steps, errors, fixes, and setup requirements encountered while making the Streamlit application work.

1. Core Components of the App

- Frontend/UI: Streamlit (app.py)
- Data Source: Google Sheets (via gspread)
- Authentication: Google Cloud Service Account (JSON credentials, later stored in st.secrets)
- Styling: Streamlit custom CSS (white background, black text, emojis, banners)
- Deployment Targets: GitHub + Streamlit Cloud (preferred)

2. Key Libraries Required

Add these to requirements.txt:

```
streamlit
pandas
gspread
google-auth
transformers
```

Optional (if using LLM summarization):
torch

3. Common Errors & Fixes

Google Service Account Credentials

Error:

Error loading sheet: [Errno 2] No such file or directory: '...json'

Cause: Hard-coded path to JSON file not found.

Fix: Store credentials in st.secrets instead of committing .json.

Correct setup:

- secrets.toml inside .streamlit/ folder:

```
[google_service_account]
type = "service_account"
project_id = "xxxx"
private_key_id = "xxxx"
private_key = "-----BEGIN PRIVATE KEY-----\nxxxx\n-----END PRIVATE
KEY-----\n"
```

```
client_email = "xxxx@xxxx.iam.gserviceaccount.com"
client_id = "xxxx"
auth_uri = "https://accounts.google.com/o/oauth2/auth"
token_uri = "https://oauth2.googleapis.com/token"
auth_provider_x509_cert_url = "https://www.googleapis.com/oauth2/v1/certs"
client_x509_cert_url = "https://www.googleapis.com/robot/v1/metadata/x509/xxxx"
```

```
- In app.py:
    from google.oauth2.service_account import Credentials

    creds =
    Credentials.from_service_account_info(st.secrets["google_service_account"])
    gc = gspread.authorize(creds)
```

GitHub Secret Scanning / Push Protection

Error:
remote: Push cannot contain secrets

Cause: Accidentally committed the Google Service Account JSON.
Fix:

1. Remove the file:
git rm --cached practical-lodge-341703-f87bbdac7e18.json
echo "*.json" >> .gitignore
2. Rewrite Git history (since the secret was committed before):
pip install git-filter-repo
git filter-repo --path practical-lodge-341703-f87bbdac7e18.json --invert-paths
3. Force push:
git push origin main --force

Invalid OAuth Scope

Error:
(`'invalid_scope: Invalid OAuth scope or ID token audience provided.'`)

Cause: Mixing up `gspread.service_account(filename=...)` with `Credentials.from_service_account_info`.
Fix: Always use the same credential method (`from_service_account_info`) with `st.secrets`.

🎨 UI/Styling Issues

- Dark Mode overriding CSS → fixed by forcing `color: black !important;`
- Table readability → applied CSS for black text + white background
- Expander not stretching → applied CSS override for `.streamlit-expander`
- Banner image → used `st.image('assets/banner.png', use_container_width=True)`
- Background image (optional):

```
st.markdown(
    f"""
    <style>
    .stApp {{
        background: url("https://path/to/image.jpg");
        background-size: cover;
    }}
    </style>
    """,
    unsafe_allow_html=True
)
```

⚡ Streamlit Caching

- Used `@st.cache_data(ttl=600)` to cache Google Sheets data for 10 minutes.
- Prevents repeated API calls.

4. Successful Implementations

- Filters: Region, Industry, Category
- Error Handling: Displays warning when invalid filters chosen
- Results Display: Emojis + styled banners
- Table: Expandable, black text on white
- Multiple Banners: Two stacked images supported
- Secure Authentication: via `st.secrets`

5. Deployment Instructions

Option A: Streamlit Cloud (Easy)

1. Push repo to GitHub (without secrets).
2. Go to Streamlit Cloud (<https://share.streamlit.io>).
3. Deploy by linking repo → branch → `app.py`.
4. Add secrets in Streamlit Cloud → App Settings → Secrets.

Option B: Local / Heroku / Docker

- Heroku requires requirements.txt + Procfile.
- Docker requires a Dockerfile.
- Run locally with:
streamlit run app.py

6. Tips for Sharing the App

- Use a public Google Sheet if you don't want users to authenticate.
- Document filters in the sidebar so users know what to expect.
- Share the Streamlit Cloud URL with collaborators.

Biggest Lessons Learned:

1. Never commit secrets (Google JSON).
2. Always use st.secrets in Streamlit for production.
3. Debugging file paths is crucial (No such file or directory often means missing /Users/...).
4. UI tweaks require forcing CSS overrides.
5. Git history must be cleaned if secrets were ever committed.