Technical Guide - Connecting PetConnect

Contents

1 System Components for PetConnect

1.1 Development Environment

1.1.1 Git

Version Control: Git is used to manage and track changes in the project's codebase. It ensures efficient collaboration and provides version history, making it easy to roll back changes if needed. The project is hosted on Streamlit Community Cloud via GitHub, with the repository URL for development collaboration: `https://github.com/orhaziza/PetConnect.git`. Branching strategies and pull requests are utilized to manage different features and bug fixes, ensuring code integrity and smooth collaboration.

1.2 Backend Technologies

1.2.1 Python

Python is responsible for backend processing, including:

- **Server-Side Logic:** Handling user interactions, API requests, and responses.
- **Google Sheets Integration:** Interacting with the Google Sheets API to read and write data related to user submissions, form data, and the system's operational data.

Libraries:

- `gspread` is used to handle reading/writing Google Sheets.
- `streamlit-gsheets-connection` simplifies the connection between Streamlit and Google Sheets.
- Other essential Python libraries like `pandas` for data manipulation and `hashlib` for security (password hashing).

Required Python Libraries:

```
bcrypt==4.0.1
matplotlib==3.9.1
numpy==1.24.4
pandas==1.5.3
streamlit==1.36.0
streamlit-aggrid==1.0.5
streamlit-extras==0.4.3
streamlit-option-menu==0.3.13
toml==0.10.2
st-gsheets-connection==0.0.4
google-cloud-storage
pyppeteer
asyncio
weasyprint==62.3
gspread
oauth2client
xhtml2pdf==0.2.16
```

1.2.2 CSS

CSS is utilized to enhance the visual styling of the backend-generated web pages. This includes:

- **Custom Styling:** Elements like buttons, tables, and containers are styled using external CSS (styles.css) to improve user experience.
- **Background Images and Fonts:** Managed through local files (e.g., background3.png) and CSS to maintain branding consistency and a pleasant user interface.

1.2.3 Database Management: Google Cloud Services (Google Sheets)

Google Sheets is used as the primary data store. All user form submissions, updates, and operational data are stored and managed via Google Sheets, enabling:

- **Real-Time Data Management:** Streamlined data access and updates.
- **Flexibility:** Easily scalable and accessible from multiple locations.

The integration is managed using Google Sheets APIs through the `gspread` library, and credentials are securely stored in the `secrets.toml` file.

1.3 API Endpoints

Read/Write operations on user-submitted form data are processed through Google Sheets API endpoints. Example: `https://sheets.googleapis.com/v4/spreadsheets/\{spreadsheetId\}/values/\{range\}`. Refer to the gspread documentation for further guidance: `https://docs.gspread.org/en/latest/user-guide.html`.

1.4 Frontend Technology

1.4.1 Streamlit

Streamlit is the primary framework used for building the frontend of the PetConnect application. Key features include:

- **Rapid UI Development:** Allows for rapid creation of web-based user interfaces without extensive frontend coding.
- **User Interaction:** Supports forms, buttons, and tables for tasks like adoption forms and profile management.
- **Data Visualizations:** Dynamic data visualization from Google Sheets (tables, charts).

1.5 Security and Configuration

1.5.1 Secrets Management (`secrets.toml`)

Sensitive information such as API keys, Google Cloud credentials, and other critical data are stored in the `secrets.toml` file. It includes:

- **Google Sheets Credentials:** client_email, private_key, and project_id for Google authentication.

2 Installation Guide for PetConnect

This guide walks through the installation and setup process for PetConnect.

2.1 Prerequisites

Before setting up the project, ensure you have:

- **Git:** Download from the official Git website.
- **Python (v3.11 recommended):** Download from the official Python website.
- **Google Cloud Project with API access:** You need a project with the Google Sheets API enabled.

2.2 Cloning the Repository

```
git clone https://github.com/orhaziza/PetConnect.git
cd PetConnect
```

2.3 Setting Up the Python Environment

- On macOS/Linux:

```
python -m venv petconnect-env
source petconnect-env/bin/activate
```

- On Windows:

```
python -m venv petconnect-env
\petconnect-env\Scripts\activate
```

Install dependencies:

```
pip install -r requirements.txt
```

2.4 Configuring Application Settings

Create `secrets.toml` under the `.streamlit` directory:

```
[gcp_service_account]
type = "service_account"
project_id = "streamlit-petconnect"
private_key_id = "5aff812d32f75dbedf4cae03c4541b332a91f08d"
private_key = "-----BEGIN PRIVATE KEY-----..."
client_email = "petconnect@streamlit-petconnect.iam.gserviceaccount.com"
client_id = "112966340493999182873"
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/petconnect%40streamlit-pet
```

Ensure Google Sheets API and Google Drive API access.

2.5 Google Sheets Setup

PetConnect uses multiple Google Sheets:

- Adoption requests spreadsheet
- Adopters spreadsheet
- Foster Homes spreadsheet
- Dogs spreadsheet
- Shopping List spreadsheet

Share spreadsheets with the service account and ensure editing permissions.

2.6 Running the Application

Start the Streamlit server:

```
streamlit run home.py
```

Access the application via:

```
http://localhost:8501
OR
https://petconnect.streamlit.app
```

2.7 Verifying the Installation

- **Login:** Check login with `admin/admin123`.
- **Google Sheets Connection:** Ensure correct data fetching.
- **Functionality:** Test form submissions and actions.

2.8 Troubleshooting Common Issues

- **Incorrect Credentials:** Check `secrets.toml`.
- **Missing Dependencies:** Run:

```
pip install -r requirements.txt
```

3 Maintenance Guide for PetConnect

3.1 Regular Updates

3.1.1 Dependencies

Regularly update Python packages and Google API libraries:

```
pip install --upgrade -r requirements.txt
```

Test updates before deploying.

3.1.2 Streamlit Updates

```
pip install --upgrade streamlit
```

3.1.3 Google Sheets API Quota Management

Monitor quotas and renew service account credentials when necessary.

3.2 Data Integrity

- **Backups:** Regularly back up Google Sheets.
- **Check for Duplicates:** Ensure no duplicate records.
- **Data Validation:** Implement validation rules.

3.3 Error Handling and Logging

3.3.1 Error Logging

Integrate logging for API failures and user authentication errors. Notify admin on critical issues.

3.3.2 API Error Management

Implement retry logic and session management to handle API failures and session expiration.

3.4 Security Best Practices

- **Secrets Management:** Keep `secrets.toml` secure and rotate credentials periodically.
- **Password Management:** Hash all passwords securely.
- **SSL:** Ensure HTTPS for data in transit.

3.5 User Training and Documentation

Provide training for non-technical users and update documentation regularly.

4 Conclusion

This guide outlines the setup, installation, and maintenance processes for PetConnect, ensuring efficient use of the platform.