This documentation will help you understand my Streamlit portfolio website code.

What is Streamlit?

Imagine you want to make a website that shows off your projects, skills, and resume, but you don't want to learn complicated web programming languages like HTML, CSS, and JavaScript from scratch. That's where **Streamlit** comes in!

Streamlit is a super cool Python library that lets you create beautiful, interactive web applications with just a few lines of Python code. It handles all the difficult web stuff behind the scenes, so you can focus on what you want to display. Think of it as a magic wand that turns your Python scripts into websites.

Understanding the portfolio_app.py Code

Let's break down my code file, portfolio_app.py, section by section.

1. Importing Streamlit and Basic Page Configuration

The first few lines of my code set up the very basic look and feel of the website.

Python

```
import streamlit as st

st.set_page_config(
    page_title="Orach Paul Francis | Portfolio",
    page_icon="🧰",
    layout="wide"
)
```

- **import streamlit as st**: This line is like saying, "Hey Python, I want to use all the cool tools that Streamlit offers, and I'll refer to them as st for short." It's a common practice to use st as an alias for streamlit.
- **st.set_page_config(...)**: This is a Streamlit function that lets you customize the browser tab settings for your website.

- **page_title="Orach Paul Francis | Portfolio"**: This sets the text that appears in the browser tab for your website. It's like the name tag for your webpage.
- **page_icon="🧰"**: This sets the small icon that appears in the browser tab next to the title. In this case, it's a toolbox emoji (🧰). You can use various emojis as icons.
- **layout="wide"**: This tells Streamlit to make your website content take up as much width as possible on the screen. The opposite is "centered," which would make the content appear in the middle with space on the sides.

2. Canva-Inspired CSS Styling

This is where my website gets its beautiful look! CSS (Cascading Style Sheets) is a language used to describe how HTML elements should be displayed. Streamlit lets you inject custom CSS using st.markdown() with unsafe_allow_html=True.

Python

```
st.markdown("""
<style>
 @import url('https://fonts.googleapis.com/css2?family=Inter:wght@400;600;800&display=swap');

 html, body, [class*="css"] {
   font-family: 'Inter', sans-serif;
   background-color: rgb(255, 255, 255);
   color: rgb(33, 33, 33);
 }

 /* ... (rest of your CSS code) ... */

</style>
""", unsafe_allow_html=True)
```

- **<style> ... </style>**: This is an HTML tag that tells the browser, "Everything inside here is CSS styling rules."
- **@import url(...)**: This line imports a special font called 'Inter' from Google Fonts. This makes your

text look consistent and professional.

- **html, body, [class*="css"] { ... }**: These lines apply styles to the entire webpage.
  - **font-family: 'Inter', sans-serif;**: Sets the default font for all text to 'Inter'. If 'Inter' isn't available, it will use a generic sans-serif font.
  - **background-color: rgb(255, 255, 255);**: Sets the background color of the entire page to white. rgb(255, 255, 255) is the code for white.
  - **color: rgb(33, 33, 33);**: Sets the default text color to a very dark grey.

Let's look at a few other important CSS rules you here:

- **.card, .stMarkdown, .stText, .stSubheader, .stTitle { ... }**: These styles are applied to specific elements that Streamlit uses or to a custom "card" class you've defined.
  - **background-color: rgb(240, 248, 245);**: Gives these elements a light greenish-white background.
  - **padding: 1rem;**: Adds some space inside the element, between the content and its border. 1rem is a unit of measurement that's relative to the font size.
  - **border-radius: 8px;**: Rounds the corners of these elements, making them look softer.
  - **box-shadow: 0 2px 6px rgba(0, 0, 0, 0.05);**: Adds a subtle shadow, making the elements appear to "pop out" a bit from the background.
- **.stButton > button { ... }**: This targets the actual button inside Streamlit's button component.
  - **background-color: rgb(0, 128, 90);**: Sets the button's background to a nice green.
  - **color: white;**: Makes the button's text white.
  - **box-shadow: 0 3px 8px rgba(0, 128, 90, 0.3);**: Adds a shadow to the button, making it look more clickable.
  - **transition: background 0.3s ease;**: Makes the color change smooth when you hover over the button.
- **.stButton > button:hover { ... }**: This applies when you move your mouse cursor over the button.
  - **background-color: rgb(0, 153, 110);**: Changes the button to a slightly lighter green when hovered, giving visual feedback.
- **h1, h2, h3 { ... }**: These style your headings (like the main title and section titles).
  - **color: rgb(0, 102, 72);**: Sets their color to a darker green.
  - **font-weight: 800;**: Makes the heading text extra bold.
- **a { ... } and a:hover { ... }**: These style your links (like those to your LinkedIn or GitHub).
  - **color: rgb(0, 128, 90);**: Sets the link color to green.
  - **text-decoration: none;**: Removes the default underline from links.
  - **text-decoration: underline;**: Adds an underline when you hover over the link.
- **hr { ... }**: This styles the horizontal rules (the dividing lines).

    - **background-color: rgb(0, 128, 90);**: Makes the line green.
- **.stImage > img { ... }**: This styles images displayed using Streamlit.
    - **border-radius: 8px;**: Rounds the corners of the image.
    - **box-shadow: 0 4px 12px rgba(0,0,0,0.1);**: Adds a shadow to the image.

unsafe_allow_html=True: This is a very important argument. It tells Streamlit, "I know what I'm doing, please allow this raw HTML and CSS to be displayed on the page." Be careful with this, as it can sometimes introduce security risks if you're not getting the HTML from a trusted source.

3. Header Section

This part of the code creates the top section of your portfolio, showing your picture and a brief introduction.

Python

```
col1, col2 = st.columns([1, 2])
with col1:
    st.image("Portfolio/orach.jpg", width=200)
with col2:
    st.title("Orach Paul Francis")
     st.markdown("#### Top Problem Solving Voice | Petroleum Engineer | Python Coder | Chairperson Membership Committee (Geological Society of Uganda) | IWCF 1 | Petrophysics | Power BI | Agile Project Management | Solutions Architecture")
    st.markdown("""
    <div class='card'>
    Dynamic and results-driven Petroleum Engineer...
    </div>
    """, unsafe_allow_html=True)
```

- **col1, col2 = st.columns([1, 2])**: This is a cool Streamlit feature that lets you create columns on your page. You're telling Streamlit to create two columns. The [1, 2] means the second column (col2) will be twice as wide as the first column (col1).
- **with col1:**: Anything indented under with col1: will be placed inside the first column.

    - **st.image("Portfolio/orach.jpg", width=200)**: This displays an image from the path "Portfolio/orach.jpg" and sets its width to 200 pixels.
- **with col2:**: Anything indented under with col2: will be placed inside the second column.
    - **st.title("Orach Paul Francis")**: This displays a large, main title for your name.
    - **st.markdown("#### ...")**: This displays text formatted using Markdown. #### means it's a level 4 heading (smaller than st.title or st.header). You use it to list your key roles and skills.
    - **st.markdown(""" ... """, unsafe_allow_html=True)**: This displays a block of text within a div element that has the card CSS class applied to it. Remember the .card styling from earlier? This makes your introductory paragraph appear in a styled box.
- **st.markdown("<hr>", unsafe_allow_html=True)**: This adds a horizontal rule (the green dividing line) across the page, separating the header from the next section.

4. Education Section

This section uses a subheader and a card to display your education details.

Python

```
st.subheader("🎓 Education")
st.markdown("""
<div class='card'>
<b>Bachelor of Science in Oil and Gas Production</b><br>
Kyambogo University
<br><br>
More available on <a href='https://linkedin.com/in/orachpaulfrancis' target='_blank'>LinkedIn</a>
</div>
""", unsafe_allow_html=True)
```

- **st.subheader("🎓 Education")**: Displays a smaller heading for this section, with a graduation cap emoji.
- **st.markdown("""...""", unsafe_allow_html=True)**: Again, you're using Markdown with embedded HTML to create a styled card for your education details.
    - **...**: Makes the text inside bold.
    - **
**: This is an HTML tag for a line break, moving the following text to the next line.

- **LinkedIn**: This is an HTML link.
  - **href='...'**: Specifies the URL the link goes to.
  - **target='_blank'**: Tells the browser to open this link in a new tab or window.

5. Featured Projects Section

This section also uses columns to display your projects side-by-side.

Python

```
st.subheader("🛠️ Featured Projects")

col3, col4 = st.columns(2)
with col3:
    st.markdown("""
    <div class='card'>
        🛢️ <b><a href='https://orach-oilfield-converter.streamlit.app/' target='_blank'>Oilfield Unit Converter</a></b><br>
    Web-based app for converting pressure, temperature, volume, flow rate, and more.
    </div>
    """, unsafe_allow_html=True)

with col4:
    st.markdown("""
    <div class='card'>
    🧙 <b><a href='https://orach.pythonanywhere.com/' target='_blank'>Sorting Hat Game</a></b><br>
    A fun web app inspired by Harry Potter's Sorting Hat — coded in Flask.
    </div>
    """, unsafe_allow_html=True)
```

- **st.subheader("🛠️ Featured Projects")**: Another subheader, this time with a tools emoji.
- **col3, col4 = st.columns(2)**: Creates two equally sized columns for your projects.
- Inside each column (with col3: and with col4:), you use st.markdown() to create a card for each project.

- Each card includes an emoji (🛢️ for oilfield, 🧙‍♂️ for magic), a **bolded link** to the project, and a short description.

6. Key Skills Section

Similar to the projects, this section also uses two columns to list your skills.

Python

```
st.subheader("🔧 Key Skills")

col5, col6 = st.columns(2)
with col5:
    st.markdown("""
    <div class='card'>
    - Log Analysis (Determin, Multimin)<br>
    - Geolog, Techlog, Petrel<br>
    - Python, Pandas, Streamlit<br>
    - ArcGIS, QGIS, Surfer<br>
    - Digital Oilfield Integration
    </div>
    """, unsafe_allow_html=True)

with col6:
    st.markdown("""
    <div class='card'>
    - Well Log Interpretation<br>
    - HSE & Field Safety Compliance<br>
    - Agile & Scrum Methodologies<br>
    - Power BI, Data Viz & Dashboards<br>
    - Team Collaboration & Project Management
    </div>
    """, unsafe_allow_html=True)
```

- **st.subheader("🔧 Key Skills")**: Subheader with a wrench emoji.
- **col5, col6 = st.columns(2)**: Two more equally sized columns.
- Within each column, st.markdown() is used to display a card with a bulleted list (-) of your skills.

7. Connect With Me Section

This section provides links to your various online profiles.

Python

```
st.subheader("🔗 Connect With Me")
st.markdown("""
<div class='card' style='line-height: 2; font-size: 16px;'>
📫 You can find me on: <br>
<a href='https://linkedin.com/in/orachpaulfrancis' target='_blank'>LinkedIn</a> |
<a href='https://github.com/ORACHPAULFRANCIS' target='_blank'>GitHub</a> |
</div>
""", unsafe_allow_html=True)
```

- **st.subheader("🔗 Connect With Me")**: Subheader with a link emoji.
- **st.markdown("""...""", unsafe_allow_html=True)**: A card containing an email emoji (📫) and a list of links to your professional and coding profiles.
    - **style='line-height: 2; font-size: 16px;'**: This is an example of **inline CSS**. It's directly applied to this specific div element.
        - line-height: 2;: Makes the spacing between lines double the normal height, making the links easier to read.
        - font-size: 16px;: Sets the font size for the text inside this card to 16 pixels.
    - The links are separated by | (pipe characters) to make them look neat.

8. Certifications Section

This is the most dynamic part of your code, where you list and display your certifications.

Python

```
st.title("📄 My Certifications")
st.markdown("To view a certificate, you will have to download it first 😹😄 ... [LinkedIn](https://linkedin.com/in/orachpaulfrancis).")

certificates = [
        {"title": "Agile Project Management", "description": "Certified in Agile principles...", "url": "https://github.com/ORACHPAULFRANCIS/.../Agile%20Project%20Management.pdf"},
    # ... (rest of your certificate data) ...
]

# Sort certificates alphabetically by title
certificates = sorted(certificates, key=lambda x: x["title"])
# Create 3 columns
cols = st.columns(3)

# Distribute certificates evenly across the 3 columns
for idx, cert in enumerate(certificates):
    with cols[idx % 3]:
        st.markdown(f"""
                <div class='card' style='padding:10px; border:1px solid #e6e6e6; border-radius:8px; margin-bottom:10px; background-color:#f9f9f9;'>
        📄 <b>{cert['title']}</b><br>
        {cert['description']}<br>
        <a href="{cert['url']}" target="_blank">View Certificate</a>
        </div>
        """, unsafe_allow_html=True)
```

- **st.title("📄 My Certifications")**: A main title for this section with a document emoji.
- **st.markdown("To view a certificate, ... [LinkedIn](https://linkedin.com/in/orachpaulfrancis).")**: A funny message explaining a "glitch" that encourages visitors to go to your LinkedIn. This adds a personal touch!
- **certificates = [...]**: This creates a Python **list** named certificates. Each item in this list is a

dictionary.

- A **dictionary** is like a small collection of related information, where each piece of information has a name (called a "key") and a value. For example, {"title": "Agile Project Management", "description": "...", "url": "..."} means:
  - The **key** is "title", and its **value** is "Agile Project Management".
  - The **key** is "description", and its **value** is a text explaining the certificate.
  - The **key** is "url", and its **value** is the link to the certificate PDF.
- This is a very organized way to store a lot of similar data.

- **certificates = sorted(certificates, key=lambda x: x["title"])**: This line sorts your list of certificates.
  - **sorted()**: This is a Python function that sorts a list.
  - **key=lambda x: x["title"]**: This is a bit advanced, but essentially it tells sorted() to sort the list based on the title of each certificate (the x represents each dictionary in the list, and x["title"] gets its title). This makes your certificates appear in alphabetical order, which is very neat!
- **cols = st.columns(3)**: This creates three equally sized columns to display your certificates. This helps arrange many items neatly on the page.
- **for idx, cert in enumerate(certificates):**: This is a **loop**! It goes through each certificate in your certificates list, one by one.
  - **enumerate()**: This is a handy function that gives you both the *index* (position, starting from 0) of the item (idx) and the *item itself* (cert) during each turn of the loop.
- **with cols[idx % 3]:**: This is where the magic of distributing certificates into columns happens.
  - **idx % 3**: The % is the "modulo" operator. It gives you the remainder after division.
    - If idx is 0, 0 % 3 is 0 (first column).
    - If idx is 1, 1 % 3 is 1 (second column).
    - If idx is 2, 2 % 3 is 2 (third column).
    - If idx is 3, 3 % 3 is 0 (back to the first column).
    - This clever trick ensures that certificates are placed in the first, then second, then third column, and then it wraps back around to the first column for the fourth certificate, and so on. This creates a balanced, grid-like layout.
- Inside the loop, for each certificate (cert):
  - **st.markdown(f"""...""", unsafe_allow_html=True)**: This uses an f-string (notice the f before the triple quotes) to insert the cert's title, description, and URL directly into the HTML string.
  - Each certificate gets its own card with a slightly different background color (background-color:#f9f9f9;) and a small border.
  - It displays the certificate title in bold, its description, and a "View Certificate" link that opens

in a new tab.

9. Footer Section

The very last part of your website is the footer.

Python

```
st.markdown("""
<hr style="margin-top: 2rem;">
<p style='text-align: center; color: grey; font-size: 0.9rem;'>
© 2025 Orach Paul Francis. Built with ❤️ using Streamlit.
</p>
""", unsafe_allow_html=True)
```

- **st.markdown("""...""", unsafe_allow_html=True)**: This displays the footer content using Markdown with embedded HTML.
  - **<hr style="margin-top: 2rem;">**: A horizontal rule at the bottom, with extra space above it.
  - **<p style='text-align: center; color: grey; font-size: 0.9rem;'>...**: A paragraph (<p>) of text.
    - **text-align: center;**: Centers the text.
    - **color: grey;**: Makes the text grey.
    - **font-size: 0.9rem;**: Makes the font slightly smaller than the default.
  - **© 2025 Orach Paul Francis. Built with ❤️ using Streamlit.**: Your copyright notice and a nice mention of Streamlit. The ❤️ is an emoji.

How to Run This Code

To see your portfolio website in action, you would typically:

1. **Save the code**: Save the code provided into a file named portfolio_app.py (or any other .py file).
2. **Open a terminal/command prompt**: Navigate to the folder where you saved portfolio_app.py.
3. **Run the command**: Type streamlit run portfolio_app.py and press Enter.

Streamlit will then open your web application in your default web browser!

Conclusion

You've built a really neat and well-structured portfolio website using Streamlit! You've used its features effectively to display your information, and the custom CSS makes it look very professional and clean.