Streamlit Notes

Part 1: UI Creation & Layout Fundamentals

Topics Covered

- Page configuration (st.set_page_config)

- Text elements: st.title, st.header, st.subheader, st.text, st.write, st.markdown, st.code

- Layout components: st.sidebar, st.columns, st.expander

Example 1: Page Setup & Text Elements (01_hello_streamlit.py)

```
import streamlit as st
# Page setup
st.set_page_config(
    page_title="Hello Streamlit",
    page_icon="👋",
    layout="centered"
)
# Titles & text
st.title("Welcome to Streamlit")
st.header("This is a Header")
st.subheader("This is a Subheader")
st.text("st.text() is used for simple fixed-width text.")
st.write("st.write() is more flexible and can display text, numbers, dataframes, etc.")
st.markdown("**st.markdown()** lets you use _Markdown_ for **rich text**.")

# Display Python code snippet
code_example = """
def add(a, b):
    return a + b
result = add(5, 7)
print(result)
"""
st.code(code_example, language="python")
```

Example 2: Sidebar, Columns, Expander (02_layout_basics.py)

```
import streamlit as st
st.set_page_config(page_title="Faculty Profile", page_icon="🎓", layout="wide")

st.title("🎓 Faculty Profile Demo")
st.markdown("This example shows how to use **sidebar**, **columns**, and **expanders**.")

# Sidebar – Important for filters/settings
st.sidebar.header("Profile Settings")
```

```
faculty_name = st.sidebar.text_input("Faculty Name", "Parth Sinroza")
department = st.sidebar.selectbox("Department", ["Computer Engineering", "IT", "AI/ML",
"CSE"])
experience = st.sidebar.slider("Years of Experience", 0, 40, 10)

st.sidebar.markdown("---")
st.sidebar.write("You can put filters, toggles, etc. in the sidebar.")

# Main content – using columns
col1, col2 = st.columns([1, 2])  # 1:2 ratio

with col1:
    st.subheader("Basic Info")
    st.write(f"**Name:** {faculty_name}")
    st.write(f"**Department:** {department}")
    st.write(f"**Experience:** {experience} years")

with col2:
    st.subheader("About")
    st.markdown("""
    Use this area to show detailed information about the faculty member,
    such as research interests, publications, and courses handled.
    """)

# Expander – for optional/extra info
with st.expander("Show Courses Handled"):
    st.write("- Data Structures")
    st.write("- Machine Learning")
    st.write("- Database Management Systems")

with st.expander("Show Publications"):
    st.write("1. Research Paper A (2021)")
    st.write("2. Research Paper B (2023)")
```

Part 2: Input Widgets & Interactivity

Topics Covered

- Text inputs: st.text_input, st.text_area

- Number inputs: st.number_input, st.slider

- Selection widgets: st.selectbox, st.multiselect, st.radio, st.checkbox

- Date/time inputs: st.date_input, st.time_input

- File uploader: st.file_uploader

- Buttons and download_button

Example 3: Text Inputs (03_text_inputs.py)

```
import streamlit as st

st.title("Text Input Demo")

name = st.text_input("Enter your name:")
comments = st.text_area("Any comments or feedback?")

st.write("**Live Output:**")
if name:
    st.write(f"Hello, **{name}** 👋")
if comments:
    st.write("Your comments:")
    st.write(comments)
```

Example 4: Number Inputs & Sliders (04_number_inputs.py)

```
import streamlit as st

st.title("Number Input & Slider Demo")

age = st.number_input("Enter your age:", min_value=0, max_value=100, value=25)
rating = st.slider("Rate this session (1–10):", min_value=1, max_value=10, value=7)

st.write(f"Your age: {age}")
st.write(f"You rated this workshop: {rating}/10")
```

Example 5: Selection Widgets (05_selection_widgets.py)

```
import streamlit as st

st.title("Selection Widgets Demo")

course = st.selectbox(
    "Select Course:",
    ["DAA", "CN", "TE-I", "EEF"]
)

preferred_days = st.multiselect(
    "Preferred Days for Extra Lectures",
    ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"]
)

delivery_mode = st.radio(
    "Preferred Delivery Mode:",
    ["Offline", "Online", "Hybrid"]
)

subscribe = st.checkbox("Subscribe to course updates?")

st.write("---")
st.write(f"**Course:** {course}")
st.write(f"**Preferred Days:** {', '.join(preferred_days) if preferred_days else 'None'}")
st.write(f"**Delivery Mode:** {delivery_mode}")
st.write(f"**Subscribed:** {'Yes' if subscribe else 'No'}")
```

Example 6: Date, Time & File Uploader (06_date_time_file.py)

```
import streamlit as st
from datetime import date, time

st.title("Date, Time & File Uploader Demo")

exam_date = st.date_input("Select Exam Date:", value=date.today())
start_time = st.time_input("Exam Start Time:", value=time(9, 0))

uploaded_file = st.file_uploader("Upload CSV file with student marks", type=["csv"])

st.write(f"Selected exam date: {exam_date}")
st.write(f"Exam start time: {start_time}")

if uploaded_file is not None:
    st.success("File uploaded successfully!")
    st.write("File name:", uploaded_file.name)
    st.write("File type:", uploaded_file.type)
```

Example 7: Buttons & Download Button (07_buttons_demo.py)

```
import streamlit as st
import pandas as pd

st.title("Button & Download Demo")

if st.button("Click to Generate Sample Marks Data"):
    df = pd.DataFrame({
        "Enrollment No": [1, 2, 3, 4],
        "Marks": [78, 85, 69, 92]
    })
    st.write("Generated Data:")
    st.dataframe(df)
    csv = df.to_csv(index=False).encode("utf-8")

    st.download_button(
        label="Download as CSV",
        data=csv,
        file_name="sample_marks.csv",
        mime="text/csv"
    )
```

Part 3: Output Display & Matplotlib Integration

Topics Covered

- Displaying data: st.dataframe, st.table, st.json
- Displaying media: st.image, st.audio, st.video
- Status elements: st.progress, st.spinner, st.success, st.error, st.warning, st.info
- Matplotlib charts: line, bar, scatter, histogram
- Streamlit built-in charts: st.line_chart, st.bar_chart, st.area_chart

Example 8: Dataframe, Table, JSON (08_display_data.py)

```
import streamlit as st
import pandas as pd

st.title("Displaying Data in Streamlit")

data = {
    "Student": ["A", "B", "C", "D"],
    "Marks": [85, 92, 76, 88],
    "Passed": [True, True, True, True]
}
df = pd.DataFrame(data)

st.subheader("st.dataframe (Interactive)")
st.dataframe(df)

st.subheader("st.table (Static)")
st.table(df)

st.subheader("st.json (Structured JSON)")
st.json(data)
```

Example 9: Media Display (09_media_demo.py)

```
import streamlit as st

st.title("Media Display Demo")

st.subheader("Image Example")
st.image(
    "Image.jpeg",
    use_container_width=True
)

st.subheader("Audio Example")      #-- “Supported Formats - .mp3, .wav, .ogg”
st.audio("Audio.mp3")

st.subheader("Video Example")      #-- “Supported Formats - .mp4, .webm, .ogv”
st.video("Video.mp4")
```

Example 10: Status Elements & Progress (10_status_demo.py)

```
import streamlit as st
import time

st.title("Status Elements Demo")

st.success("This is a success message.")
st.warning("This is a warning message.")
st.error("This is an error message.")
st.info("Useful information can go here.")

st.write("---")
st.subheader("Progress & Spinner Example")

if st.button("Start Long Task"):
    progress = st.progress(0)
    with st.spinner("Processing..."):
        for i in range(100):
            time.sleep(0.03)
            progress.progress(i + 1)
    st.success("Task completed!")
```

Example 11: Basic Matplotlib Charts (11_matplotlib_basic.py)

```
import streamlit as st
import matplotlib.pyplot as plt
import numpy as np
```

```
st.title("Matplotlib + Streamlit Demo (plt version)")

# Sample data
x = np.arange(1, 11)
y = np.random.randint(50, 100, size=10)

# ----------------------------
# Line Chart
# ----------------------------
st.subheader("Line Chart (Matplotlib)")

plt.figure(figsize=(6, 4))
plt.plot(x, y, marker="o")
plt.xlabel("Student Index")
plt.ylabel("Marks")
plt.title("Marks of 10 Students")

st.pyplot(plt)

# Clear the figure to avoid overlap
plt.clf()

# ----------------------------
# Bar Chart
# ----------------------------
st.subheader("Bar Chart (Matplotlib)")

plt.figure(figsize=(6, 4))
plt.bar(x, y)
plt.xlabel("Student Index")
plt.ylabel("Marks")
plt.title("Marks Bar Chart")

st.pyplot(plt)

# Clear again
plt.clf()
```

Example 12: Scatter, Histogram & Built-in Charts (12_matplotlib_vs_builtin.py)

```
import streamlit as st
import matplotlib.pyplot as plt
import numpy as np

st.title("Matplotlib vs Streamlit Built-in Charts (No Pandas)")

# ----------------------------
# Generate sample dataset
# ----------------------------
```

```
np.random.seed(42)
marks = np.random.normal(loc=70, scale=10, size=100)
attendance = np.random.uniform(60, 100, size=100)

# ----------------------------
# Histogram using Matplotlib
# ----------------------------
st.subheader("Histogram of Marks (Matplotlib)")

plt.figure(figsize=(6, 4))
plt.hist(marks, bins=10)
plt.xlabel("Marks")
plt.ylabel("Frequency")
plt.title("Distribution of Marks")

st.pyplot(plt)
plt.clf()

# ----------------------------
# Scatter Plot using Matplotlib
# ----------------------------
st.subheader("Scatter Plot (Matplotlib) – Marks vs Attendance")

plt.figure(figsize=(6, 4))
plt.scatter(attendance, marks)
plt.xlabel("Attendance (%)")
plt.ylabel("Marks")
plt.title("Marks vs Attendance")

st.pyplot(plt)
plt.clf()

# ----------------------------
# Streamlit Built-In Charts
# ----------------------------
st.subheader("Streamlit Built-in Line & Area Charts")

st.write("Using NumPy arrays directly:")

# Convert arrays into a simple 2D structure
chart_data = np.column_stack((marks, attendance))

st.line_chart(chart_data)
st.area_chart(chart_data)

# ----------------------------
# Streamlit Bar Chart (Histogram Approximation)
# ----------------------------
```

```
st.subheader("Bar Chart (Streamlit) – Histogram Approximation")

counts, bins = np.histogram(marks, bins=10)

st.bar_chart(counts)
```