

Unit - V

- 1) What is Streamlit,
- 2) Features of Streamlit,
- 3) Text and Table elements –
 - a. Text Elements, Titles, Headers, Subheaders ,markdowns,
 - b. Tables,
 - c. Dataframes,
- 4) Buttons and sliders –
 - a. Buttons,
 - b. RadioButton,
 - c. Checkbox,
 - d. Dropdown ,
 - e. Multiselect,
 - f. Progress bar,
 - g. Slidder,
- 5) Forms,
- 6) Develop an Streamlit Application

1) What is Streamlit,

What is Streamlit?

- ✓ **Streamlit** is an open-source Python library designed to enable data scientists and machine learning engineers to rapidly create and deploy interactive web applications for data visualization, machine learning, and analytics with minimal coding effort.
- ✓ Streamlit is an open-source Python library that allows data scientists and machine learning engineers to quickly build and share interactive web applications for their work.
- ✓ It focuses on simplicity and leverages the power of Python, eliminating the need for extensive front-end development knowledge.

Key Features of Streamlit:

- ✓ **Simplicity:** Intuitive API; write Python, get a web app.
- ✓ **Python-Native:** Leverages familiar Python syntax and data structures.
- ✓ **Real-time Updates:** Automatic app refresh on script changes.
- ✓ **Interactive Widgets:** Buttons, sliders, dropdowns, etc., for user input.
- ✓ **Automatic UI Rendering:** Smart rendering of DataFrames, plots, and other data.
- ✓ **Caching (@st.cache_data, @st.cache_resource):** Optimizes performance by memoizing function results.
- ✓ **Layout and Organization:** Columns, expanders, tabs for structuring apps.
- ✓ **Theming:** Customizable appearance.
- ✓ **Community & Ecosystem:** Active community and growing number of components.
- ✓ **Deployment:** Easy deployment on various platforms.

Text and Table Elements

These elements are used to display textual information and tabular data.

a. Text Elements, Titles, Headers, Subheaders, markdowns,

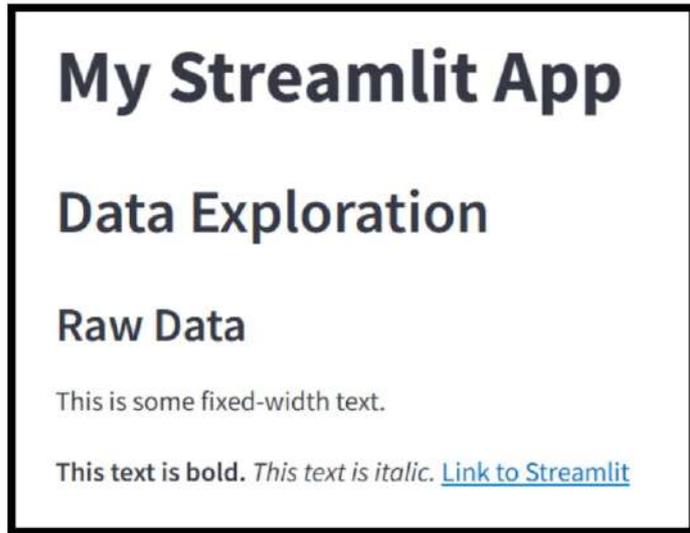
- ✓ **st.title(body):** Displays the main title of your application.
- ✓ **st.header(body):** Displays a section header, typically smaller than the title.
- ✓ **st.subheader(body):** Displays a subsection header, smaller than the header.
- ✓ **st.text(body):** Displays fixed-width, preformatted text. Useful for displaying code or raw output.
- ✓ **st.markdown(body):** Displays text formatted using Markdown syntax. This allows for rich text formatting (bold, italics, lists, links, etc.).

✓ Example

```
1 import streamlit as st
2 st.title("My Streamlit App")
3 st.header("Data Exploration")
4 st.subheader("Raw Data")
5 st.text("This is some fixed-width text.")
6 st.markdown("**This text is bold.** *This text is italic.* [Link to Streamlit](https://streamlit.io)")
```

✓

✓ Output



b. Tables,

- ✓ **st.table(data=None)**: Displays a static table.
- ✓ **data** (*Pandas.DataFrame, dict, list*): The data to display in the table. Streamlit infers column names and data types.
- ✓ It's useful for displaying small, fixed datasets or summaries.
- ✓ Streamlit handles the rendering of the table.
- ✓ can pass a Pandas DataFrame or a dictionary-like object.

c. **st.dataframe(data=None, height=None, width=None)**: Displays an interactive table (Pandas DataFrame). Offers sorting, searching, and resizing.

- ✓ **data** (*Pandas.DataFrame*): The DataFrame to display.
- ✓ **height** (*int or None*, default None): Fixed height of the dataframe in pixels.
- ✓ **width** (*int or None*, default None): Fixed width of the dataframe in pixels.
- ✓ Example

```
1 import streamlit as st
2 import pandas as pd
3 st.title("To Illustrate Streamlit Components")
4 df = pd.read_csv('emp.csv')
5 st.subheader("Static Table created through st.table")
6 st.table(df)
7 st.subheader("Interactive Table created through st.dataframe")
8 st.dataframe(df)
```

To Illustrate Streamlit Components

Static Table created through st.table

	eno	ename	desg	sal	deptno
0	e0001	smith	sales Person	20000	10
1	e0002	Jones	Manager	45000	20
2	e0003	King	Analyst	30000	10
3	e0004	krishna	Data Engineer	40000	10
4	e0005	khan	Analyst	35000	20

Interactive Table created through st.dataframe

	eno	ename	desg	sal	deptno
↑ Sort ascending		smith	sales Person	20000	10
↓ Sort descending		Jones	Manager	45000	20
↔ Autosize		King	Analyst	30000	10
📌 Pin column		krishna	Data Engineer	40000	10
🔕 Hide column		khan	Analyst	35000	20

4) Buttons and sliders

a. Buttons - Creates a Clickable Button

st.button(label, key=None, on_click=None, args=None, kwargs=None, disabled=False).

- ✓ label (*str*): The text displayed on the button.
- ✓ key (*str* or *None*, default *None*): A unique key for the widget, used for state management.
- ✓ on_click (*callable* or *None*, default *None*): A function to call when the button is clicked.
- ✓ args (*tuple* or *None*, default *None*): Positional arguments to pass to the on_click function.
- ✓ kwargs (*dict* or *None*, default *None*): Keyword arguments to pass to the on_click function.
- ✓ disabled (*bool*, default *False*): Whether the button is disabled.

```
if st.button("Click Me"):  
    st.write("Button was clicked!")
```

Click Me

Button was clicked!

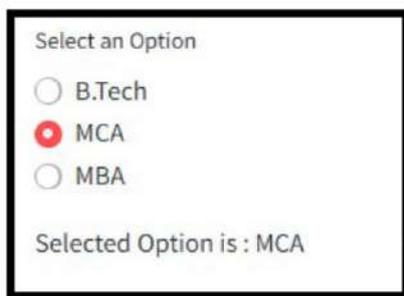
::

b. Radio Button - Creates a set of Mutually Exclusive Options

`st.radio(label, options, index=0, key=None, horizontal=False, disabled=False, on_change=None, args=None, kwargs=None)`:

- ✓ `label (str)`: The label displayed above the radio buttons.
- ✓ `options (sequence)`: A sequence of options to display.
- ✓ `index (int, default 0)`: The index of the option to select by default.
- ✓ `key, disabled, on_change, args, kwargs`: Same as `st.button`.
- ✓ `horizontal (bool, default False)`: Whether to display options horizontally.

```
branch = st.radio("Select an Option", ["B.Tech", "MCA", "MBA"])
st.write(f"Selected Option is : {branch}")
```



c. Check Box - Creates a Single CheckBox

`st.checkbox(label, value=False, key=None, disabled=False, on_change=None, args=None, kwargs=None)`:

- ✓ `label (str)`: The label displayed next to the checkbox.
- ✓ `value (bool, default False)`: The initial state of the checkbox (checked or unchecked).
- ✓ `key, disabled, on_change, args, kwargs`: Same as `st.button`.

```
checked = st.checkbox("I agree")
if checked:
    st.write("You agreed!")
```



d. Drop Down - Creates a dropdown list for single selection

`st.selectbox(label, options, index=0, key=None, disabled=False, on_change=None, args=None, kwargs=None)`:

- ✓ `label (str)`: The label displayed above the select box.
- ✓ `options (sequence)`: A sequence of options to display in the dropdown.

- ✓ `index (int, default 0)`: The index of the option to select by default.
- ✓ `key, disabled, on_change, args, kwargs`: Same as `st.button`.

```
selected_item = st.selectbox("Choose B.Tech Branch:", ["CSE", "EEE", "ECE", "AIML", "DS"])
st.write(f"You chose: {selected_item}")
```

e. **multiselect** - Creates a dropdown list for multiple selections

`st.multiselect(label, options, default=None, key=None, disabled=False, on_change=None, args=None, kwargs=None)`:

- ✓ `label (str)`: The label displayed above the multiselect box.
- ✓ `options (sequence)`: A sequence of options to display.
- ✓ `default (sequence or None, default None)`: A sequence of options to select by default.
- ✓ `key, disabled, on_change, args, kwargs`: Same as `st.button`.

```
selected_multiple = st.multiselect("Choose Your Multiple Hobbies:", ["Reading", "Painting", "Dancing", "Sports"], default=["Reading"])
st.write(f"You selected: {selected_multiple}")
```

f. **Progress Bar** - Displays a Progress Bar

`st.progress(value, text='')`.

- ✓ `value (float or int)`: The progress value, between 0 and 1 (or an integer representing percentage).
- ✓ `text (str)`: Optional text to display alongside the progress bar.

```
import time
my_bar = st.progress(0)
for i in range(101):
    time.sleep(0.01)
    my_bar.progress(i / 100)
st.write("Progress complete!")
```



Progress complete!

g. Slider: Creates a slider for selecting a numerical value

- **st.slider(label, min_value=None, max_value=None, value=None, step=None, format=None, key=None, disabled=False, on_change=None, args=None, kwargs=None):**
 - ✓ label (*str*): The label displayed above the slider.
 - ✓ min_value (*int, float, datetime.date, datetime.datetime* or *None*, default *None*): The minimum value of the slider.
 - ✓ max_value (*int, float, datetime.date, datetime.datetime* or *None*, default *None*): The maximum value of the slider.
 - ✓ value (*int, float, datetime.date, datetime.datetime, tuple* of two such values, or *None*, default *None*): The initial value(s) of the slider. Can be a single value or a tuple for a range slider.
 - ✓ step (*int, float, datetime.timedelta* or *None*, default *None*): The step size of the slider.
 - ✓ format (*str* or *None*, default *None*): A format string to display the slider value (e.g., "%.2f").
 - ✓ key, disabled, on_change, args, kwargs: Same as st.button.

```
age = st.slider("Select your age:", 0, 100, 25)
st.write(f"Your age is: {age}")

price_range = st.slider("Select a price range:", 0.0, 100.0, (25.0, 75.0))
st.write(f"Selected price range: {price_range}")
```

Select your age:

0 100

25

Your age is: 25

Select a price range:

0.00 100.00

25.00 75.00

Selected price range: (25.0, 75.0)

5) Forms

- **with st.form(key='my_form'): ...:** Creates a form that groups widgets.
 - key (str): A unique key for the form.
- **st.form_submit_button(label='Submit', on_click=None, args=None, kwargs=None, help=None, disabled=False):** Creates a submit button within a form.
 - ✓ label (str): The text displayed on the submit button.
 - ✓ on_click, args, kwargs, disabled: Same as st.button.
 - ✓ help (str or None, default None): A tooltip displayed when hovering over the button.

```

1 import streamlit as st
2 import pandas as pd
3 import numpy as np
4
5 with st.form(key='student_form'):
6     st.subheader("Personal Information")
7     name = st.text_input("Full Name", placeholder="Enter your full name")
8     email = st.text_input("Email Address", placeholder="Enter your email")
9     age_slider = st.slider("Select your age", 0, 100, 25)
10    st.write(f"Your age (from slider): {age_slider}")
11    gender = st.radio("Gender", ["Male", "Female"])
12    date_of_birth = st.date_input("Date of Birth")
13    st.subheader("UG DETAILS")
14    degree = st.selectbox("Degree", ["B.Sc", "BCA", "BCOM", "Other"])
15    specialization = st.selectbox("specialization in UG", ["Statistics", "Electronics", "Computer Science"])
16    per = st.text_input("UG Percentage", placeholder="Enter your UG Percentage")
17    st.subheader("Contact Information")
18    phone_number = st.text_input("Phone Number", placeholder="Enter your phone number")
19    address = st.text_area("Address", placeholder="Enter your current address")
20    st.subheader("Additional Information (Optional)")
21    extracurricular_activities = st.text_area("Extracurricular Activities", placeholder="List any extracurricular activities")
22    submit_button = st.form_submit_button("Submit Application")
23    if submit_button:
24        st.success("Application submitted successfully!")
25        st.balloons()

```

Personal Information

Full Name

Ivan

Email Address

ivan@gmail.com

Select your age



Your age (from slider): 25

Gender

- Male
 Female

Date of Birth

2015/05/05

UG DETAILS

Degree

B.Sc

specification in UG

Statistics

UG Percentage

75

Contact Information

Phone Number

9786757678

Address

Chittoor

Additional Information (Optional)

Extracurricular Activities

Cricket, Hockey

Submit Application

Application submitted successfully!

R.P. Admaja