

## **Streamlit – Subjective Test**

**Name:- Kush Milkesh Mistry**

**ID:- MST03 – 0049**

Q1) Ans:- Streamlit is an open source app framework, that allows developers, with limited web development experience, to create and share apps of Machine Learning and Data Science projects.

Main Features:-

- 1) Easy to use:- Streamlit is easy to use, even for those with no prior experience in web apps. We can make apps by writing only a few lines of python code.
- 2) Real time interactivity:- We can make interactive apps with Streamlit, by using widgets such as sliders, buttons, and text inputs.
- 3) Automatic Reload:- Whenever our source code is saved, Streamlit reruns the whole code, thus showing the updates made in our code. It makes iterative development faster.
- 4) Data Caching:- Streamlit can store the results of costly computations, so that they need not be executed on every rerun.

Q2) Ans:-

- Streamlit is specifically designed for data – driven apps, whereas Flask and Django are general purpose web app frameworks.
- Streamlit is easier to deploy with built-in cloud hosting options, but apps made by Flask and Django need more configuration for deployment.
- Streamlit has built in support for real-time interactivity, like widgets, but in Flask and Django, we need additional JavaScript for that purpose.
- Streamlit is integrated with various libraries such as Numpy, Pandas, etc. which is useful for data analysis and visualization, but with Flask and Django, we need additional tools.
- Apps in Streamlit can be made with few lines of python code, but in Flask and Django, we need more setup and boilerplate code.

Q3) Ans:- Typical Use cases of Streamlit:-

- Data analysis Dashboards:- Interactive Dashboards for exploring and visualizing data
- Prototyping:- Quickly building and iterating on prototypes for data applications.
- ML model demos:- Showcasing and testing ML models.
- Reporting tools:- Creating dynamic reports with real time data visualization.

Q4) Ans:- We can create a simple streamlit app by importing streamlit module( if not installed:- then installing it by pip install streamlit) in our python program, and adding few lines of code for frontend, by calling various functions in streamlit module( using st. ) like st.title(), st.image(), etc. and then running it on our terminal by streamlit run filename.py

Q5) Ans:-Basic structure of a streamlit script:-

```
import streamlit as st
```

```
"""
```

Our code (includes both code for backend functionality in python, and for frontend by calling functions of streamlit)

```
"""
```

#Example:-

```
st.title(".....")
```

```
...
```

```
....
```

```
....
```

Q6) Ans:-

- For adding sliders:- use `st.slider("label_for_slider")`
- For adding buttons:- use `st.button("Text_inside_button")`
- For adding text inputs:- use `st.text_input("label")`

Q7) Ans:- For user interaction, Streamlit provides widgets such as sliders, buttons, and text inputs, allowing users to interact with app in real time.

For state management, Streamlit has Session State, which stores variables which have user-specific data. It basically manages and maintains the state of these variables in the app.

Q8) Ans:-Best practices:-

- User-friendly interface:- Design a user-friendly interface with clear visualization and intuitive navigation.
- Modular Code:- Break your code in reusable functions and modules.
- Dependency Management:- Include a `requirements.txt` file to manage dependencies.
- State Management:- Use session state to handle user-specific data, and maintain app state throughout interactions.
- Performance Optimization:- use caching (i.e. `@st.cache_data`) for storing the results of expensive computations to improve performance.

Q9) Ans:- To deploy a streamlit app locally, use the command:- `streamlit run filename.py`

on the terminal. We must make sure that we are in the same directory where the script is. After typing the command and pressing Enter, we will be taken to our web browser, where we can see our web app.

Q10) Ans:- To deploy a streamlit app, follow these steps:

- Firstly, host your app on github.
- Then go to Streamlit Sharing, login into your account( or create one, if not already and connect it with github)
- There, click on Create app-> I have one app already-> paste your Github URL-> Deploy.

Q11) Ans:- The requirements.txt file mentions the list of all Python libraries and their versions, on which the app depends, thus ensuring that these packages are installed in our deployment environment in Streamlit. It should be inside the same directory as the script(in github).