1.What is Streamlit and what are its main features?

It is an open-source framework used in creating applications. It can be used for deployment. It is free to use, and we don’t need to have prior web development knowledge while creating an application.

2.How does Streamlit differ from other web application frameworks like Flask or Django?

- Streamlit is especially used for data science and for ml projects but the other two can be used for various purposes.
- To build an application, using Streamlit is far better because we can create it with minimal code. Speed is high in Streamlit when compared to Flask or Django.
- Deployment can be done in simple way in Streamlit. But in other frameworks it requires more configuration and setup.

3.What are some typical use cases for Streamlit?

- Data Analysis
- ML model Deployment
- Business Intelligence
- Interactive Reports
- Scientific Research

4.How do you create a simple Streamlit app?

- Install Streamlit: pip install streamlit.
- Create a Python script
- Import Streamlit: import streamlit as st.
- Use Streamlit functions to build the app:
- st.title("My Streamlit App")
- st.write("Hi, Streamlit!")
- Run the app: streamlit run app.py.

5.Can you explain the basic structure of a Streamlit script?

Basic Structure of a Streamlit Script

1. Import Libraries:
2. Set Up the App Layout:
3. Add Interactive Elements:
4. Display Data

5. Run the App

6.How do you add widgets like sliders, buttons, and text inputs to a Streamlit app?

Use Streamlit’s built-in widget functions to add interactivity. Examples include:

```
python
import streamlit as st

for Slider:
value = st.slider("Select a value", 0, 100)

for Button:
if st.button("Click me"):
    st.write("Button clicked!")

for Text Input:
text = st.text_input("Enter text")
```

7.How does Streamlit handle user interaction and state management?

Streamlit reruns the script from start to finish with every user interaction. To maintain state across these reruns, Streamlit uses st.session_state, allowing values to persist and be updated as users interact with the app.

8.What are some best practices for organizing and structuring a Streamlit project?

- Modularize code by dividing it into functions and modules.
- Use clear, descriptive names for files and variables.
- Keep the main script focused on layout and interactions.
- Employ version control to manage changes.
- Add documentation and comments for better understanding and maintenance.

9.How would you deploy a Streamlit app locally?

- Ensure all required dependencies are installed.
- Run the app with the command: streamlit run app.py.
- Access the app via the local server URL provided by Streamlit

10.Can you describe the steps to deploy a Streamlit app?

- Prepare the app and list all dependencies in requirements.txt.
  For Streamlit Cloud:
  - Push your code to a GitHub repository.
  - Connect your GitHub account on Streamlit Cloud and deploy the repository.

11.What is the purpose of the requirements.txt file in the context of Streamlit deployment?

The requirements.txt file lists all the Python dependencies needed to run the Streamlit app. During deployment, this file ensures that all necessary packages are installed, allowing the app to run correctly in the deployment environment.