**Streamlit Theoretical Questions**

**Streamlit - Deployment Module**

**Subjective Test**

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

**Answer:** The streamlit is nothing but the open source that build for python and Python library that makes it easy to create and share custom web applications for machine learning and data science.

The main features of streamlit is:

Simple to use:

it is easy to use streamlit, the structure of streamlit is easy to understand and to create a custom web application.

Real time interaction:

It take less time to create web application and Streamlit apps automatically update in real-time as you modify the code or interact with widgets. This makes it easy to prototype and iterate on your app quickly.

Helps to build Web applications:

As we early say it’s a open source which helps us to build web application and it can be done as fast as it can, Streamlit apps can be deployed easily using Streamlit Sharing, a free cloud service, or other cloud platforms like Heroku, AWS, GCP, and Azure.

Customization and layout:

Streamlit provide a lots of Varity of layout that can easily be customize and change as we want, You can create columns, tabs, and expandable sections to organize content.

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

**Answer:** Streamlit, Flask, and Django are all popular frameworks for creating web applications, but they have distinct purposes and features that set them apart.

Streamlit shines in rapid development and deployment for data-driven applications, Flask offers flexibility and simplicity for custom web services, and Django provides a comprehensive framework for building scalable, feature-rich web applications.

**3.What are some typical use cases for Streamlit?**

### Answer: There are various typical use for streamlit like,Report Generation Creating automated reports that combine text, data tables, and visualizations into a single, interactive document.

### **Data Collection and Annotation** Streamlit can be used to build tools for collecting and annotating data, providing a simple interface for users to input or label data.

### **Data Science Prototyping** Streamlit allows data scientists to quickly prototype data analysis and machine learning workflows, sharing results with colleagues or stakeholders without needing complex web development.

Machine Learning Model Deployment Streamlit simplifies the process of deploying machine learning models. You can create an interactive app where users can input data and get predictions or classifications from a trained model in real-time.

**4.How do you create a simple Streamlit app?**

**Answer:** Install straemlit, you can use the pip for the installation,

create a python script that will be for streamlit web application, and save it by (.py)

After that import the library,

and add some text , inputs and buttons

Display data chart, run it,

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

**Answer:** Streamlit allows you to write app just like you write python code, it Perform any necessary data loading, cleaning, or processing based on user input. Create and display visualizations based on the processed data. Show the results or outputs of the data analysis or model predictions,

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

**Answer:** Streamlit built-in widgets is to enable user interaction.

Slider

number = st.slider('Pick a no', min\_value=0, max\_value=100, value=50)

st.write(f'The selected number is {number}')

Button

if st.button('Click me'):

st.write('Button clicked!')

else:

st.write('Button not clicked yet.')

Text input

user\_input = st.text\_input('Enter some text')

st.write(f'You entered: {user\_input}')

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

**Answer:**whenSession State Streamlit provides a way to persist information across reruns using st.session\_state.

st.session\_state is a dictionary-like object that allows you to store variables and retain their values across reruns.

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

**Answer:** Use a version control system like git and github to track changes and collaborate with others.

Add comments and docstrings to explain the purpose of functions and modulesBreak down the app into separate modules and functions to avoid cluttering the main script. And organizing the directory

**9.How would you deploy a Streamlit app locally?**

**Answer:**

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

**Answer:**

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

**Answer:** The requirements.txt file is for used to list all the Python dependencies required by your Streamlit app.

It ensures that all necessary packages are installed in the correct versions when deploying the app to a server or sharing it with others. This file is can be difficult for consistent environments and smooth deployment.