Module 8: Local Storage and Persistence

Theory Assignments:

1. Difference Between Local Storage Options (shared_preferences, SQLite, Hive):

- **shared_preferences**: Stores small amounts of simple data in key-value pairs (like settings or preferences). It's fast and easy but not meant for large or complex data.
- **SQLite**: A relational database that stores structured data in tables (like rows and columns). It's suitable for handling complex, large datasets that require querying.
- **Hive**: A fast, lightweight, NoSQL database for Flutter, designed for storing data in key-value pairs. It's great for large datasets and objects and provides higher performance than SQLite in some cases.

2. CRUD Operations and How They Are Implemented in SQLite or Hive:

CRUD Operations:

- **Create**: Add new data.
- **Read**: Retrieve data.
- **Update**: Modify existing data.
- **Delete**: Remove data.

In SQLite:

- **Create**: Use SQL INSERT to add records.
- **Read**: Use SQL SELECT to get records.
- **Update**: Use SQL UPDATE to change data.
- **Delete**: Use SQL DELETE to remove records.

In Hive:

- **Create**: Use box.put() to insert data.
- **Read**: Use box.get() to retrieve data.
- **Update**: Use box.put() to overwrite data.
- **Delete**: Use box.delete() to remove data.

3. Advantages and Use Cases for shared_preferences:

Advantages:

- **Simple**: Very easy to use and integrate.
- **Fast**: Great for small amounts of data like settings or flags.
- **Lightweight**: Doesn’t require a complex setup, just key-value pairs.

Use Cases:

- **Settings**: Store user preferences like theme (dark/light mode).
- **Flags**: Track app states, like whether the user has completed onboarding.
- **Session Info**: Store small authentication tokens or IDs for the current session.