Assignment Summary

Approach

1. **Planning and Requirements Gathering:**
    - Identified the core functionalities: user login/logout, service management (add, view, and delete services), and persistence of services across sessions.
    - Chose `SharedPreferences` for local storage and `Provider` for state management to ensure efficient and scalable app architecture.
2. **UI/UX Design:**
    - Designed separate screens for Admin and User roles, ensuring a responsive layout using `MediaQuery` for adaptability to various screen sizes.
    - Integrated an intuitive user interface with features like floating action buttons, dialogs for adding new services, and snackbars for user feedback.
3. **Feature Implementation:**
    - **Login and Logout:**
        - Utilized `SharedPreferences` to track user login status and redirect to appropriate pages.
    - **Service Management:**
        - Implemented CRUD operations for services.
        - Used `Provider` for updating the service list dynamically across the app.
    - **Persistence:**
        - Stored and retrieved services as a JSON-encoded string in `SharedPreferences`.
    - **Navigation:**
        - Implemented navigation between screens using `Navigator.push` and `Navigator.pushReplacement`.
4. **Testing and Iteration:**
    - Ensured that all features worked seamlessly through rigorous testing on multiple devices and emulators.
    - Focused on edge cases, such as empty service names, and handled them appropriately.

Challenges Faced

1. **Responsive Design:**
    - Ensuring the app's layout adapts to different screen sizes was challenging.
    - **Solution:** Used `MediaQuery` to dynamically calculate dimensions, padding, and font sizes.
2. **State Management:**
    - Maintaining a consistent state across widgets while handling dynamic updates.
    - **Solution:** Utilized the `Provider` package to simplify state management and ensure smooth updates to the UI.
3. **Persistent Storage:**

    - Converting between data types (e.g., List and JSON) for storage and retrieval in `SharedPreferences`.
    - **Solution:** Leveraged the `jsonEncode` and `jsonDecode` functions for serialization and deserialization of data.
4. **Testing and Debugging:**
    - Debugging issues related to asynchronous data loading.
    - **Solution:** Used `async` and `await` to ensure proper synchronization.

Testing Methodology

1. **Functional Testing:**
    - Verified that each feature (login/logout, adding/deleting services, navigation) worked as expected.
2. **UI/UX Testing:**
    - Tested responsiveness on devices with different screen sizes and resolutions.
    - Ensured user interface consistency and smooth interactions.
3. **Edge Case Testing:**
    - Checked for scenarios like adding an empty service name, deleting the last service, and handling invalid inputs..