Flutter App Development with SQLite and Django Integration

1. Objective

The objective of this assignment is to develop a Flutter application with integration of SQLite database for local storage and Django REST API for backend services. The app will allow users to record product entries, view product details, delete products, and display notifications for successful actions.

2. Requirements

a. Splash Screen with Lottie Animation

- Display a splash screen with a Lottie animation upon app launch.
- Configure the animation properly within the app.

b. Home Page:

- Implement an app bar with the following features:
- Title dynamically fetched from SQLite DB table via Django API, displaying the number of rows in the table.
- Leading icon: Close button to exit the app, triggering an alert box upon click.
- Trailing icon: Dropdown list of products fetched from SQLite table via Django REST API, showing only product names.
- Include text fields for Name, Product ID (readonly, randomly generated), and Description with appropriate validations.
- Disable the submit button if fields are not validated.
- Display a success notification upon successful submission of data.
- Ensure data submission to SQLite DB via Django API.

c. Update App Bar and List of Products:

- Update the app bar title and list of products dynamically whenever a new product is added.

d. Details Screen for Products:

- Implement a details screen for products, accessible by clicking on a product in the dropdown list.
- Display product details including name, ID, and description.
- Provide an option to delete the product.
- Show an alert box for delete confirmation.
- Perform the delete action in SQLite DB via Django API.
- Navigate back to the home screen after deletion, updating the displayed data.

3. Implementation Guidelines:

- Utilize Flutter packages for updating data in the app bar.

- Define routes for navigation within the app.
- Make the app bar a global widget for consistency.
- Use a global scaffold for screens to maintain a consistent layout.
- Manage state effectively throughout the app for efficient data handling.

4. Submission Guidelines

- Submit the Flutter project source code along with any necessary documentation.
- Ensure proper organization of code files and adherence to Flutter coding standards.
- Include clear comments and explanations where necessary to facilitate understanding.
- Submit the assignment before the specified deadline.

5. Evaluation Criteria

- Adherence to the provided requirements and specifications.
- Functional implementation of all features with proper error handling.
- Code quality, readability, and maintainability.
- Effective usage of Flutter packages and integration with SQLite and Django API.
- Compliance with Flutter coding standards and best practices.

6. Submission Deadline

24 hours

7. Note

Feel free to reach out for clarification on any aspect of the assignment. Good luck!