LOGIN SCREEN

Exercise #01

Flutter Developer Bootcamp

# **Purpose**

This Exercise represents a simple login screen UI with text input fields for username and password, login button, and options for social login or signing up.

# **Problem**

The Flutter Exercise creates a login screen UI with username and password text fields, a login button, social media login options, and a signup link. It uses Material Design widgets like Scaffold, Container, TextField, ElevatedButton, IconButton, Row, and TextButton to structure the UI.

# **How to Solve**

1. Checkout the workshop from Git Repo:

git clone -b <user-branch> <repo-URL>

2. Open the root folder inside VS Code

3. To build the app click the run option in the main method { }

4. Create a login screen with:

* Two textfield
* A elevated button
* some icons
* A text and textbutton

# **You Will Achieve**

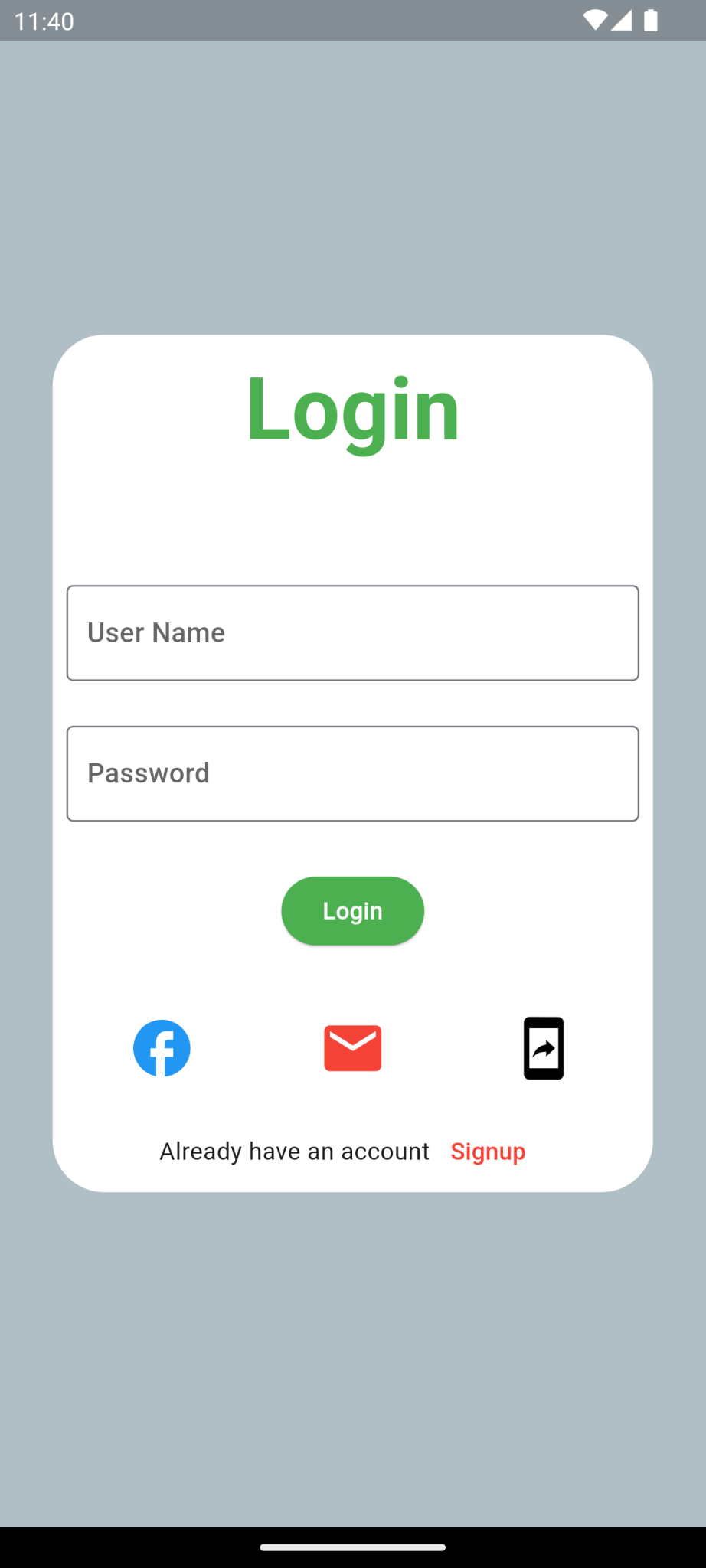
Your Flutter code creates a simple login screen UI. Here's a breakdown of what it achieves:

1. **UI Layout**: It defines a UI layout for a login screen using Flutter's Scaffold widget, Container, Column, Text, TextField, ElevatedButton, IconButton, and TextButton.
2. **Styling**: It applies styling to various UI elements using properties like backgroundColor, height, width, decoration, borderRadius, style, and colour.
3. **Interactivity**: It includes interactive elements like ElevatedButton, IconButton, and TextButton, though the onPressed callbacks are currently empty (() {}), so they don't perform any actions.
4. **Social Login**: It provides options for social login using IconButton with icons for Facebook, email, and mobile.
5. **Routing**: It includes a "Signup" button which presumably would lead to another screen for user registration, although the onPressed callback is currently empty.
6. **Accessibility**: It uses const keywords wherever possible to optimise performance by avoiding unnecessary rebuilds.

Overall, it's a well-structured Flutter code for a basic login screen UI, but you might want to implement functionality for the buttons and add error handling for text inputs.

# **Screenshots**

## **Expected Output**



# **How to submit your Exercise**

Push your project back to the same git branch using command:

<command name>

# **Happy Coding!**