Edit page with Regex validation

Exercise #1

Flutter Developer Bootcamp

# **Purpose**

This exercise demonstrates how to implement a user-friendly account editing feature in a Flutter application, covering various UI elements, form validation using regular expressions, navigation, and user interaction aspects.

**Problem**

In the provided Exercise You need to create as follows:

In the given exercise Create a Edit Page for users to edit their account details. It's implemented as a StatefulWidget managing form state. The screen features input fields for name, email, phone number, and address, with validation logic for email and phone number. Users can save their changes using an ElevatedButton.

**How to Solve**

1. Checkout the exercise 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. Initialize TextEditingController: Initialize TextEditingController instances for managing text input.

5. TextEditingController \_emailController = TextEditingController();

6. TextEditingController \_phoneNumberController = TextEditingController();

7. Build UI: Override the build() method to define the UI for the Edit5 screen using a Scaffold widget.

8. Define Form and Input Fields: Inside the SingleChildScrollView widget, wrap the form content with a Form widget and define input fields using TextFormField.

9. Apply Validation: Implement validation logic for email and phone number fields inside TextFormField widgets.

10. Go To File: <specific-file-with-validation-method> à <method-name>, implement your logic.

**You will Achieve**

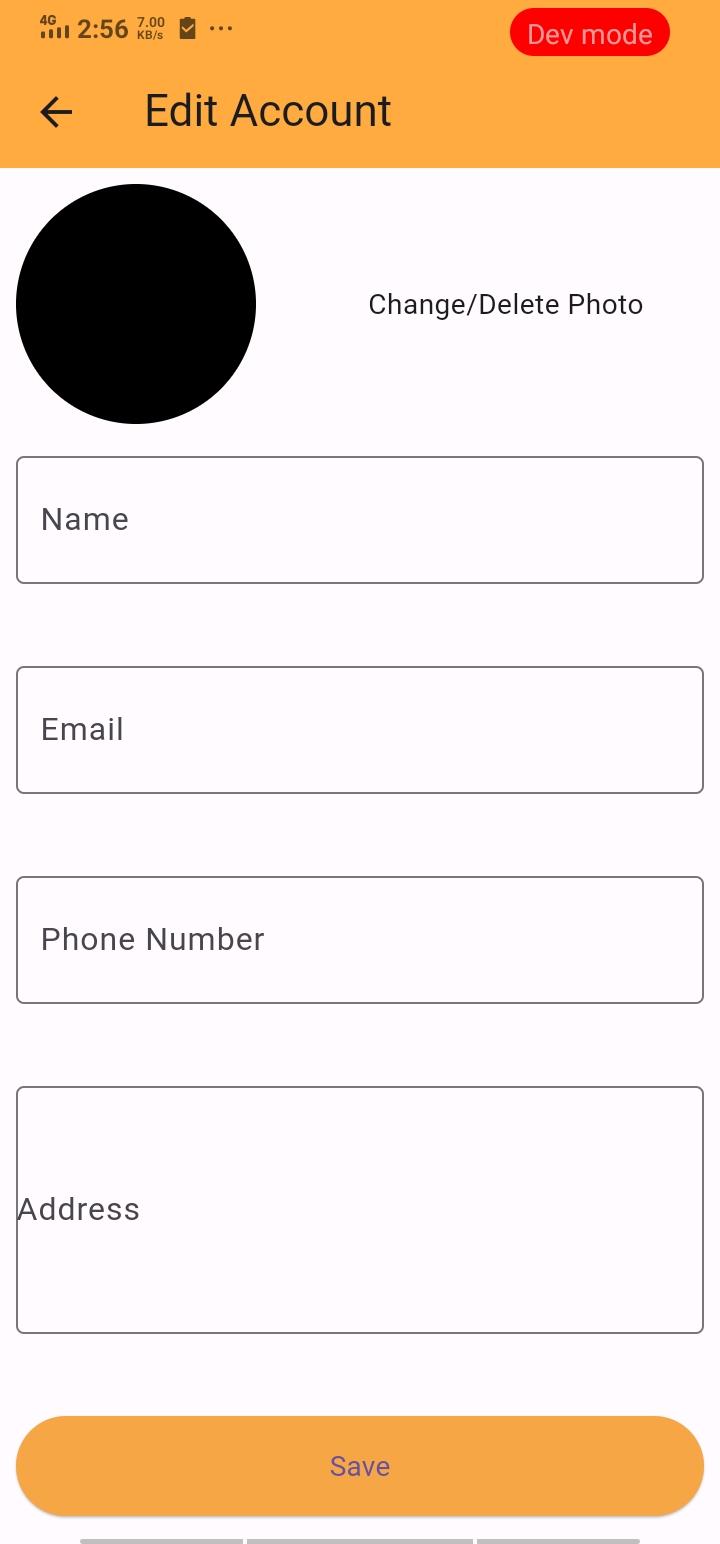
When you complete this Exercise you will learn the following:

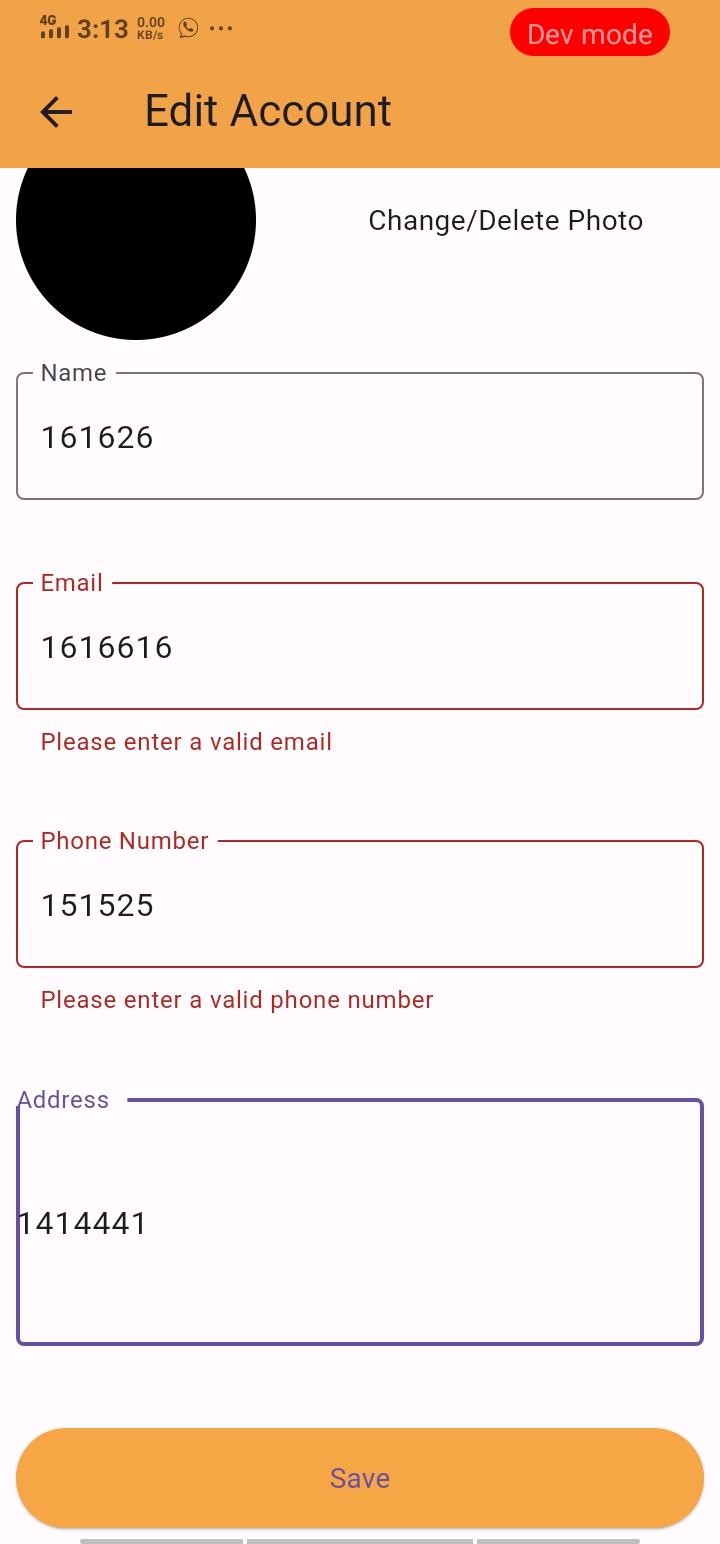
**Widgets Included:**

* **Scaffold:** The overall structure of the screen, which provides app bars, drawers, and other common UI elements.
* **AppBar:** A material design app bar with a title "Edit Account", a custom orange accent color, and a back button (IconButton) on the leading side.
* **SingleChildScrollView:** Wraps the entire form to enable scrolling when the keyboard is displayed to prevent overflow issues.
* **Form:** A container for form fields that enables form validation, submission, and management. It uses a global key (\_formKey) to identify the form.
* **Row: Contains two child widgets:** a CircleAvatar and a Text widget. This row is for displaying the user's profile picture and an option to change or delete the photo.
* **Padding:** Used to add space around the form fields and buttons.
* **TextFormField:** Input fields for user information such as name, email, phone number, and address. Each TextFormField is wrapped in Padding to add spacing. The email and phone number fields have validators for input validation.
* **ElevatedButton:** A button used to save the changes made to the account details. It triggers the \_saveForm() method when pressed.
* **\_saveForm():** A method called when the form is submitted (i.e., when the save button is pressed). Currently, it only prints "Form submitted!" to the console but could be expanded to handle form submission logic.

# **Screenshots**

## **Expected output(Edit page with regex validation)**





# **How to submit your Exercise**

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

<command name>

# **Happy Coding!**