

CROSS-PLATFORM MOBILE APP DEVELOPMENT

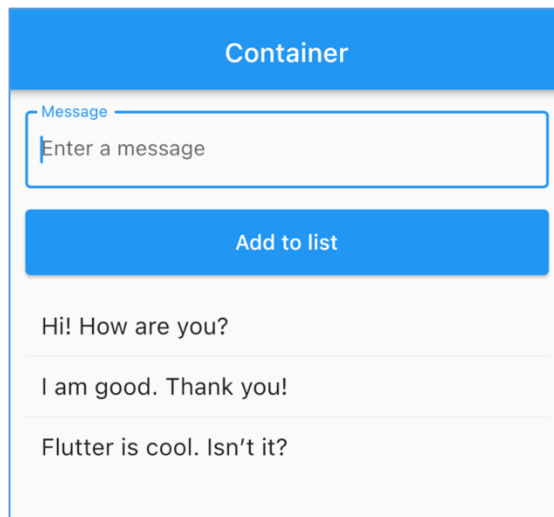
(503107)

LAB 5

EXERCISE 1

Create a basic Flutter app that allows the user to enter text into a TextField widget then add the text to a ListView widget underneath.

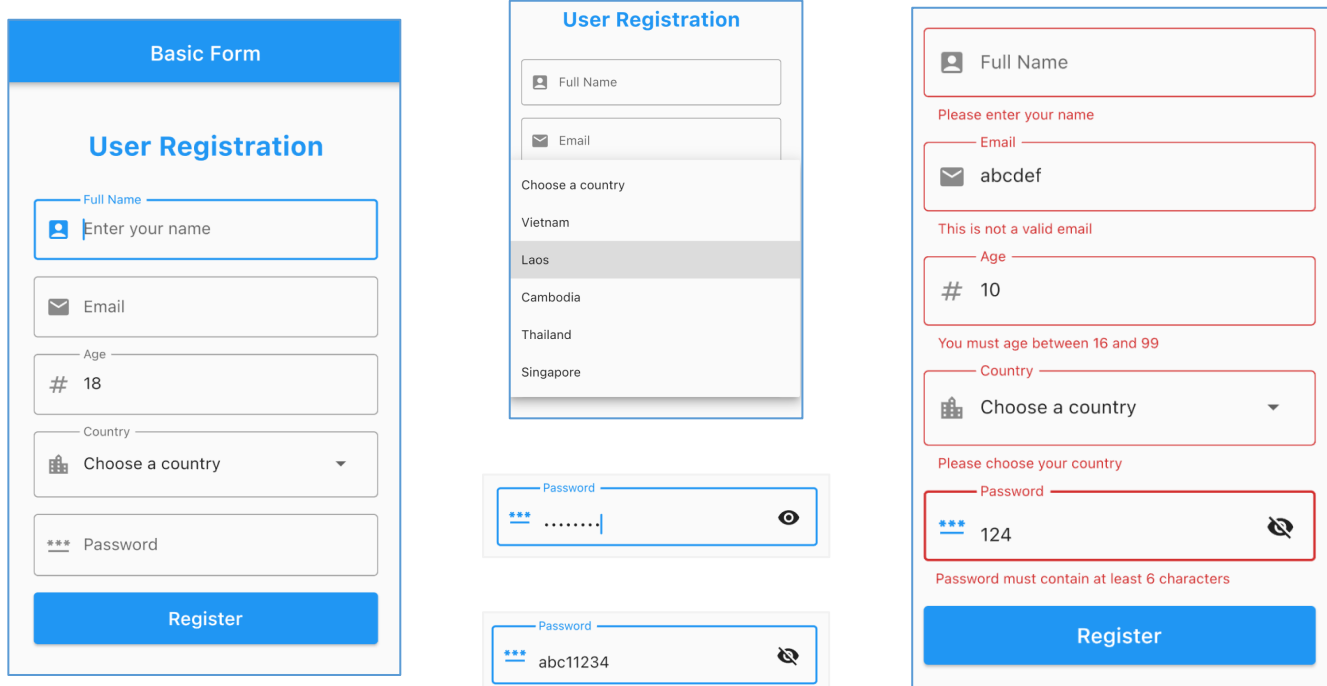
1. You need to check and display the appropriate error message with the Snackbar widget if the user leaves the content blank.
2. When successfully adding an item to the listview, it is necessary to empty the textfield and give the cursor focus automatically to the textfield so that the user can continue to enter the value quickly.
3. In addition, users can also press the Enter key to add content instead of clicking the button.



The screenshot shows a mobile app interface within a blue container. At the top, there is a text input field with the placeholder text 'Enter a message' and a blue border. Below the input field is a blue button with the text 'Add to list'. Underneath the button is a list of three messages: 'Hi! How are you?', 'I am good. Thank you!', and 'Flutter is cool. Isn't it?'. Each message is on a separate line with a light gray background and a thin blue border.

EXERCISE 2

Create an application that displays the user registration screen using the Form and the TextFormField, DropdownButtonFormField widgets. When selecting fields, the appropriate keyboard should be displayed. For example, you need to display the numeric keypad when entering your age, the keyboard with the @ sign when entering an email address. You also need to set up appropriate input validation for each field. For example, the age field only allows input of up to 2 digits, the username and password must be at least 6 characters, the email value must contain the @ sign... Error messages will disappear immediately after the user the user has entered a valid value without waiting for the button to be pressed.



The image shows three variations of a 'User Registration' form:

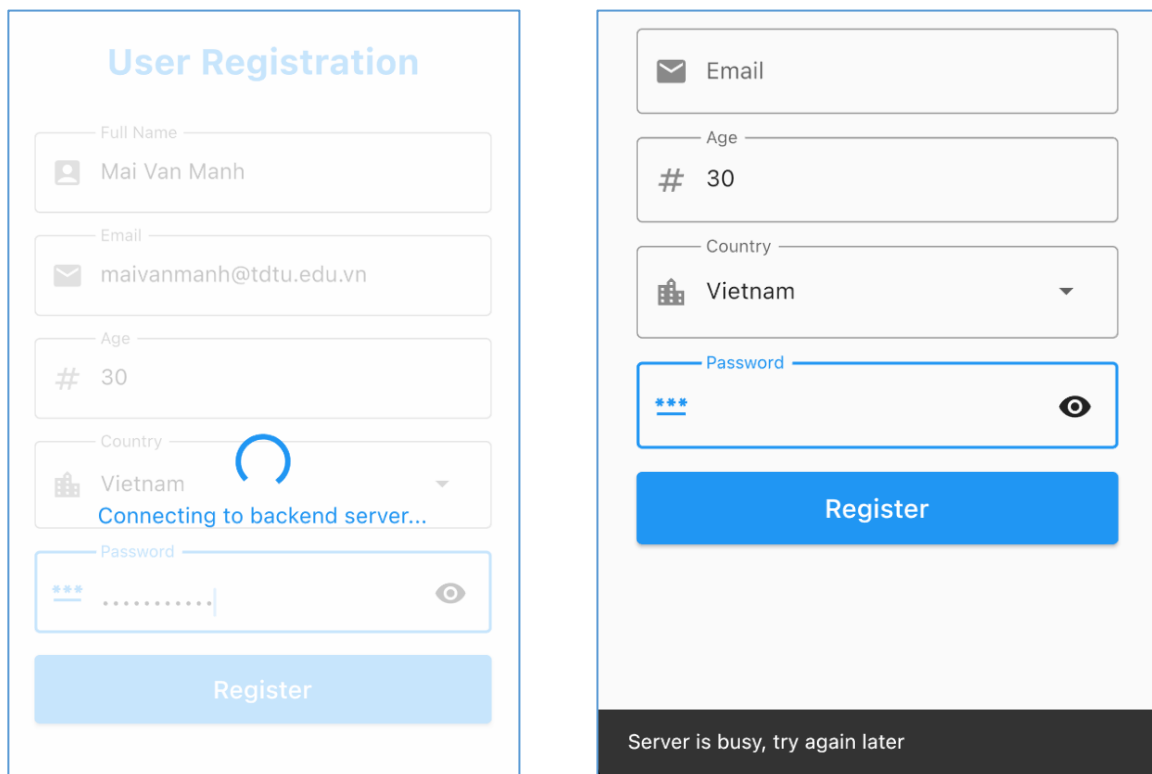
- Left Screenshot:** The initial form with fields for Full Name, Email, Age, Country (dropdown), Password, and a Register button.
- Middle Screenshot:** The form with the Country dropdown menu open, showing a list of countries: Vietnam, Laos, Cambodia, Thailand, and Singapore.
- Right Screenshot:** The form with validation errors. The Full Name field has the error 'Please enter your name'. The Email field has the error 'This is not a valid email' and contains the text 'abcdef'. The Age field has the error 'You must age between 16 and 99' and contains the text '# 10'. The Password field has the error 'Password must contain at least 6 characters' and contains the text '124'.

With the country field, the user is not allowed to enter the value arbitrarily, but can only be selected from the available list. The password field should provide a suffix widget that allows the user to toggle between masking and unmasking the password.

If the user has filled in all valid values, then print them to the console, delete all the content in the form and set automatic focus for the name field.

EXERCISE 3

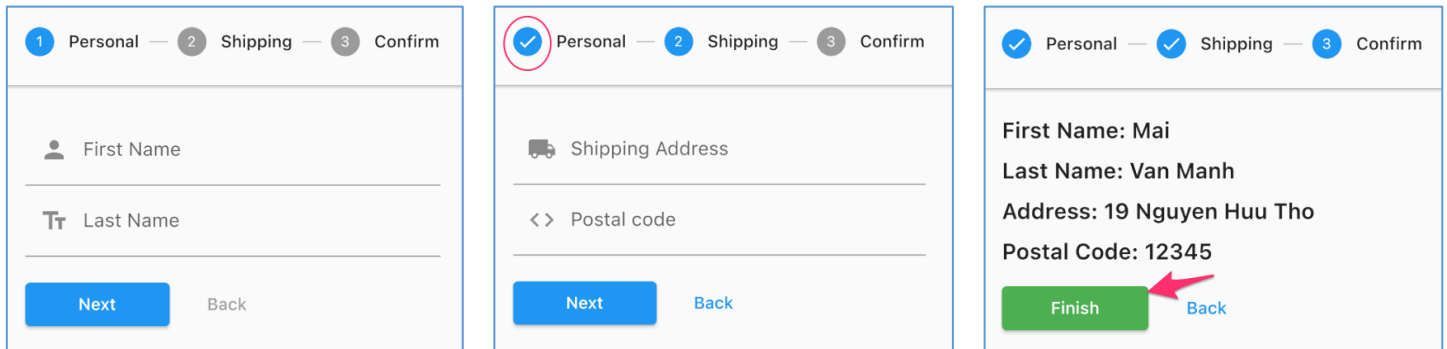
Adjust the previous exercise to simulate sending data to the server when performing account registration. On successful form submission, display a `CircularProgressIndicator` and status text in the center of the screen, and prevent the user from interacting with the form during this period. After 3s, hide the loading screen and display the message "server is busy, try again later" with `Snackbar` widget.



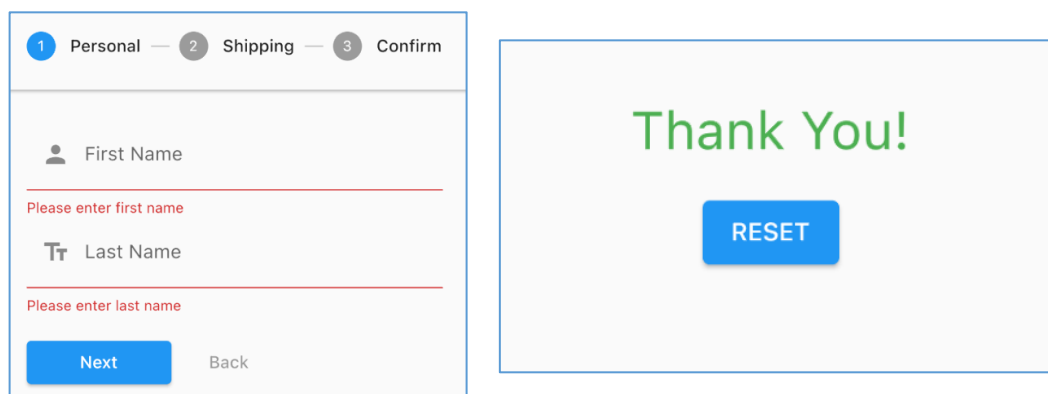
The image displays two states of a mobile application's user registration screen. The left state shows the registration form with fields for Full Name (Mai Van Manh), Email (maivanmanh@tdtu.edu.vn), Age (30), Country (Vietnam), and Password. A blue circular progress indicator is overlaid on the Country field, and the text "Connecting to backend server..." is displayed below it. The right state shows the same form, but the progress indicator is hidden, and a blue "Register" button is visible. At the bottom of the screen, a black Snackbar displays the message "Server is busy, try again later".

EXERCISE 4

Use the `Stepper` widget to create a multi-step form application that has 3 steps: Personal Info, Shipping Address and Confirm screen. In the first two steps, users need to enter enough and correct information to pass the next steps.



When moving to a new step, the old step will be marked as complete with a check mark. In the last step, the button will change the background color to green. At each step, the user can use the next to move to the next step (if they have entered all the required information), the user can also press the Back button or tap on the title of the previous steps to go back to that step.



In the first step, of course, the user cannot press back. In the last step, after pressing finish, the user will see the word "Thank you" and a reset button to be able to start from the beginning.