CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY

FACULTY OF TECHNOLOGY AND ENGINEERING

CHANDUBHAI S. PATEL INSTITUTE OF TECHNOLOGY

DEPARTMENT OF ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

Subject : Mobile Application Development   **Semester:** 5

Subject Code: AIML308   **Academic Year :**2025-26(ODD)

NAME : CHOKSI OM CHIRAGBHAI   **ID: 23AIML010**

Practical 4

Problem Definition

You are building a mobile application where users navigate through multiple screens like login, dashboard, and profile. Design a Form-based Registration App with validation using TextFormField.

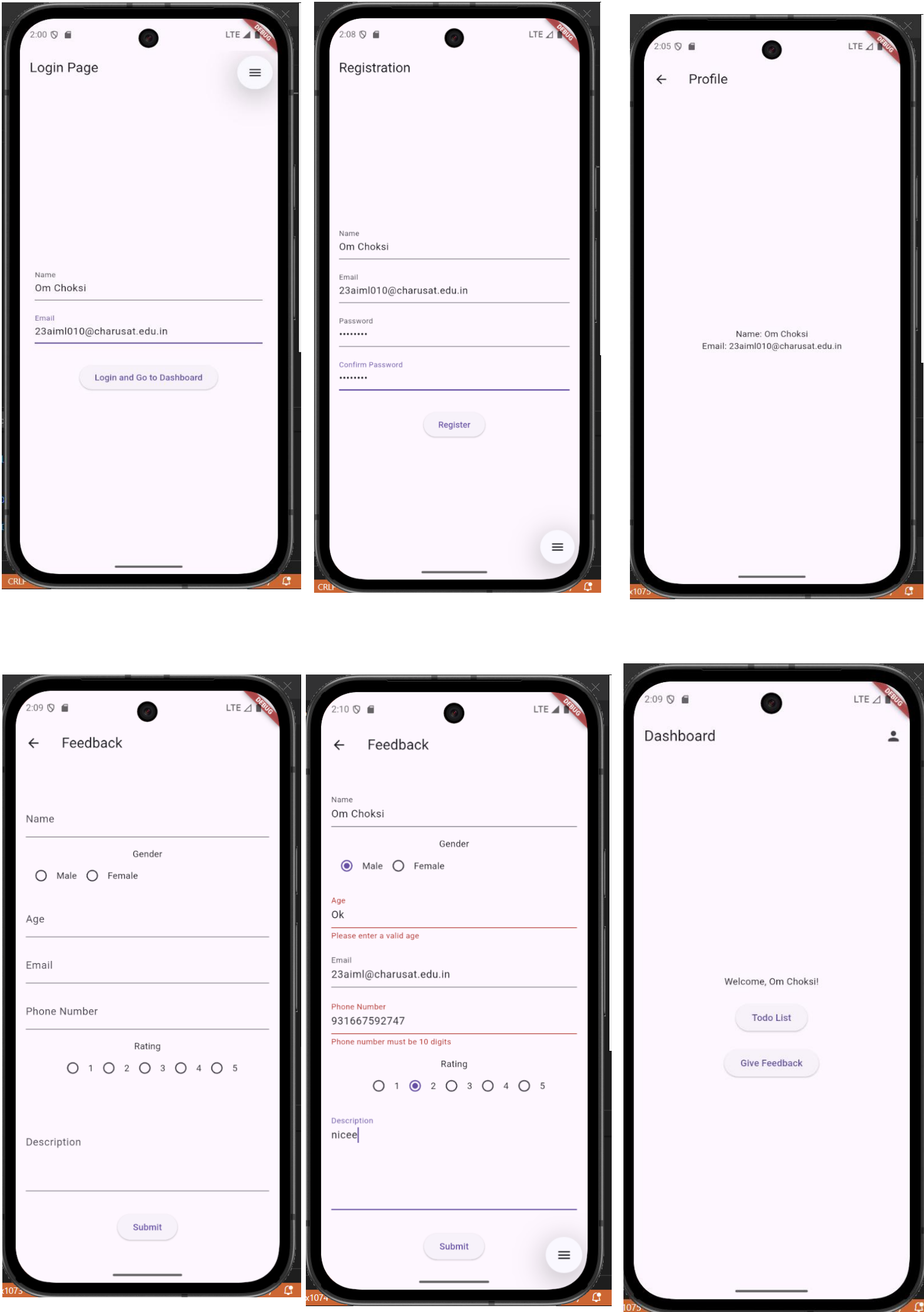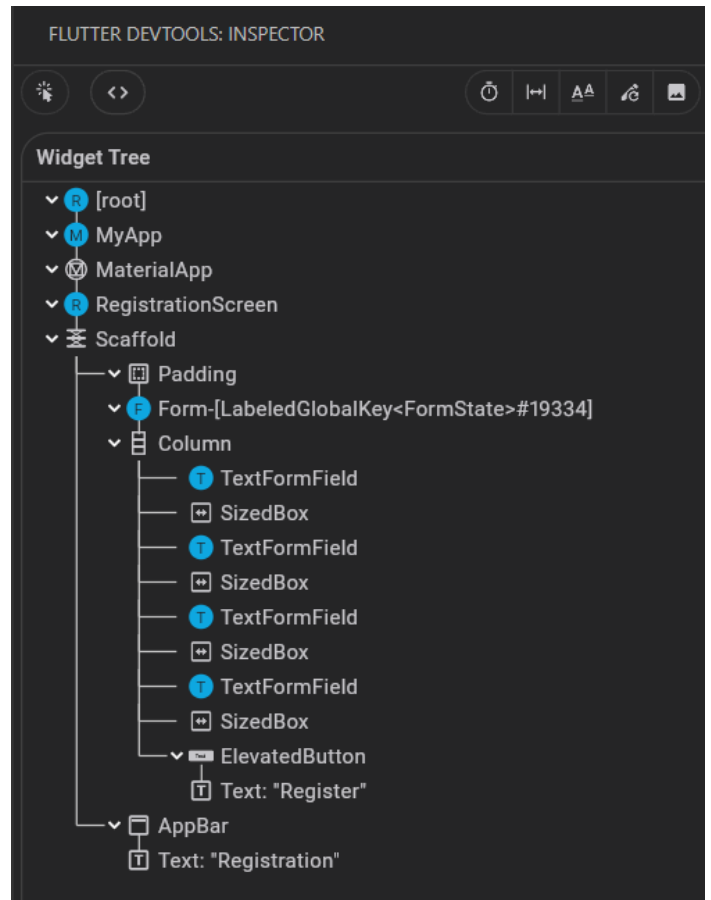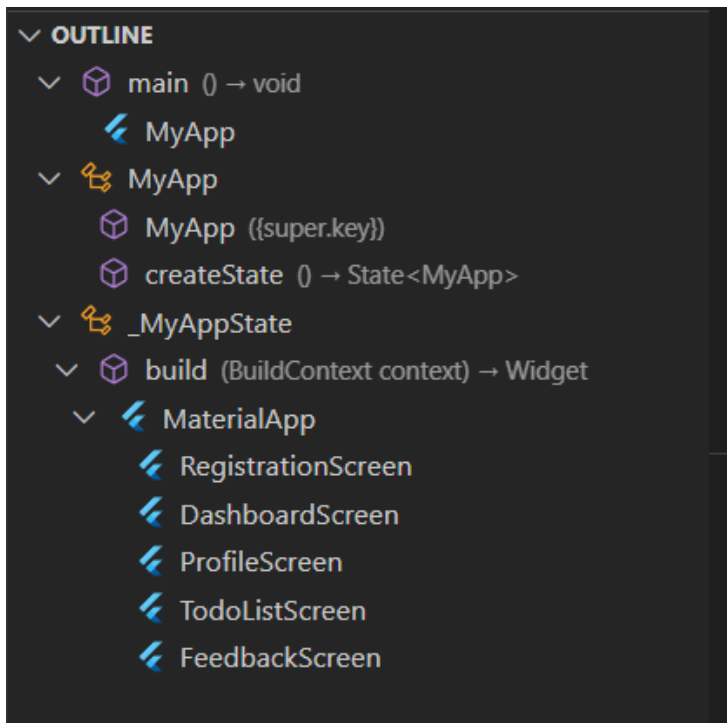| Supplementary Problems -	Feedback form with rating

Technical Approach:

This Flutter application implements a form-based registration system with validation, utilizing TextFormField for input handling and GlobalKey<FormState> for form state management. The core logic employs TextEditingController for accessing input values and custom validator functions to enforce data integrity. Key widgets include Form, TextFormField, Radio, and ElevatedButton, with navigation managed through named routes in MaterialApp.

File Structure:

lib/
├── main.dart   #he entry point of the application.
├── login.dart #Handles user authentication and login logic.
├── dashboard.dart # Displays the main dashboard after login.
├── profile.dart #Manages user profile information and settings.
├── todolist.dart # Implements a to-do list feature for task management.
└── feedback.dart #Allows users to provide feedback or submit reviews.

Screenshots:

Key Questions:

1. How to take and validate user input?
Ans: Used Form widget with GlobalKey<FormState>, wrap TextFormField in Form, define validator functions that return error strings or null, call _formKey.currentState!.validate() on submit to trigger validation and display errors.

2. What is the role of Dart functions?
Ans: Create TextEditingController instances for each TextFormField to access input values, use GlobalKey<FormState> assigned to Form widget to manage form state and trigger validation programmatically.

3. How to update UI on user action?
Ans: Required field validation (check if empty), email format validation using regex, numeric validation with int.tryParse(), password strength (minimum length), field matching (confirm password), and custom length validation (e.g., phone number digits).

Key Skills to be Understand:

V Form Handling, Input Validation