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 9

Problem Definition

You are building a mobile application where users navigate through multiple screens like login, dashboard, and profile. Develop a Login Authentication App using API-based credential check and session handling.

| Supplementary Problems -	Two-factor login or role-based access

Technical Approach:

This Flutter application implements a multi-screen news app with REST API integration, focusing on asynchronous data handling and JSON parsing. The core implementation uses the http package for API calls, flutter_dotenv for secure environment variable management, and FutureBuilder for reactive UI updates based on async operations. Key widgets include StatefulWidget for state management, FutureBuilder for handling loading/error/success states, ListView.builder for dynamic article rendering, and Scaffold for consistent app structure across screens.
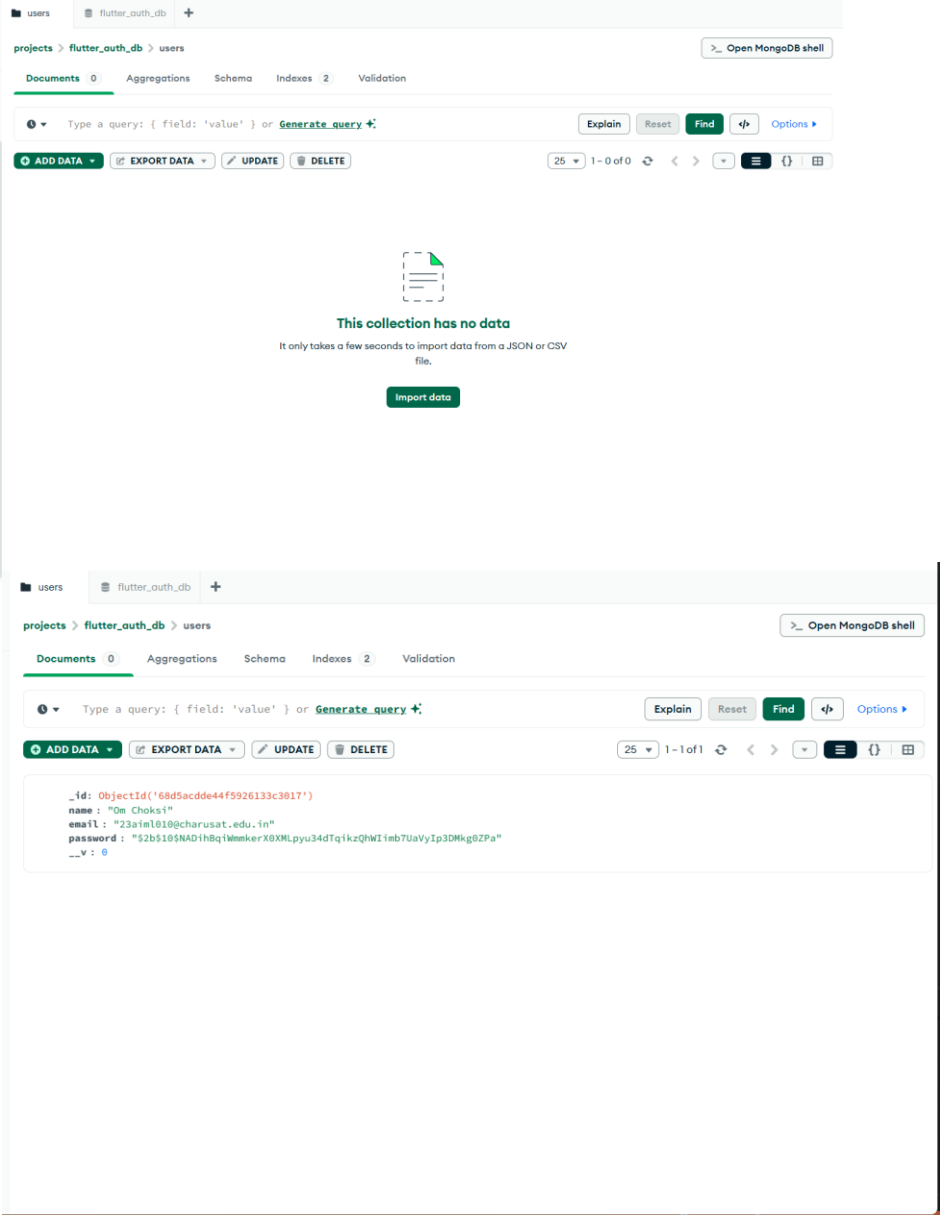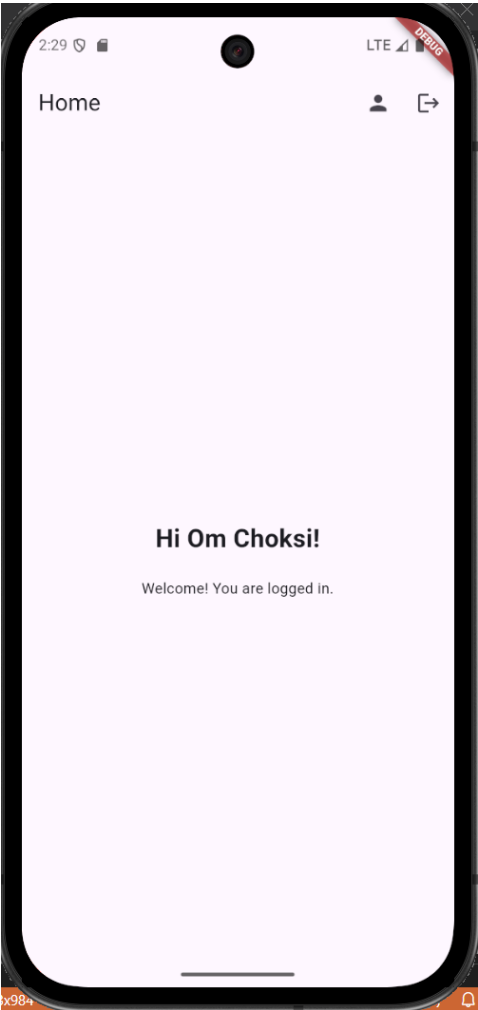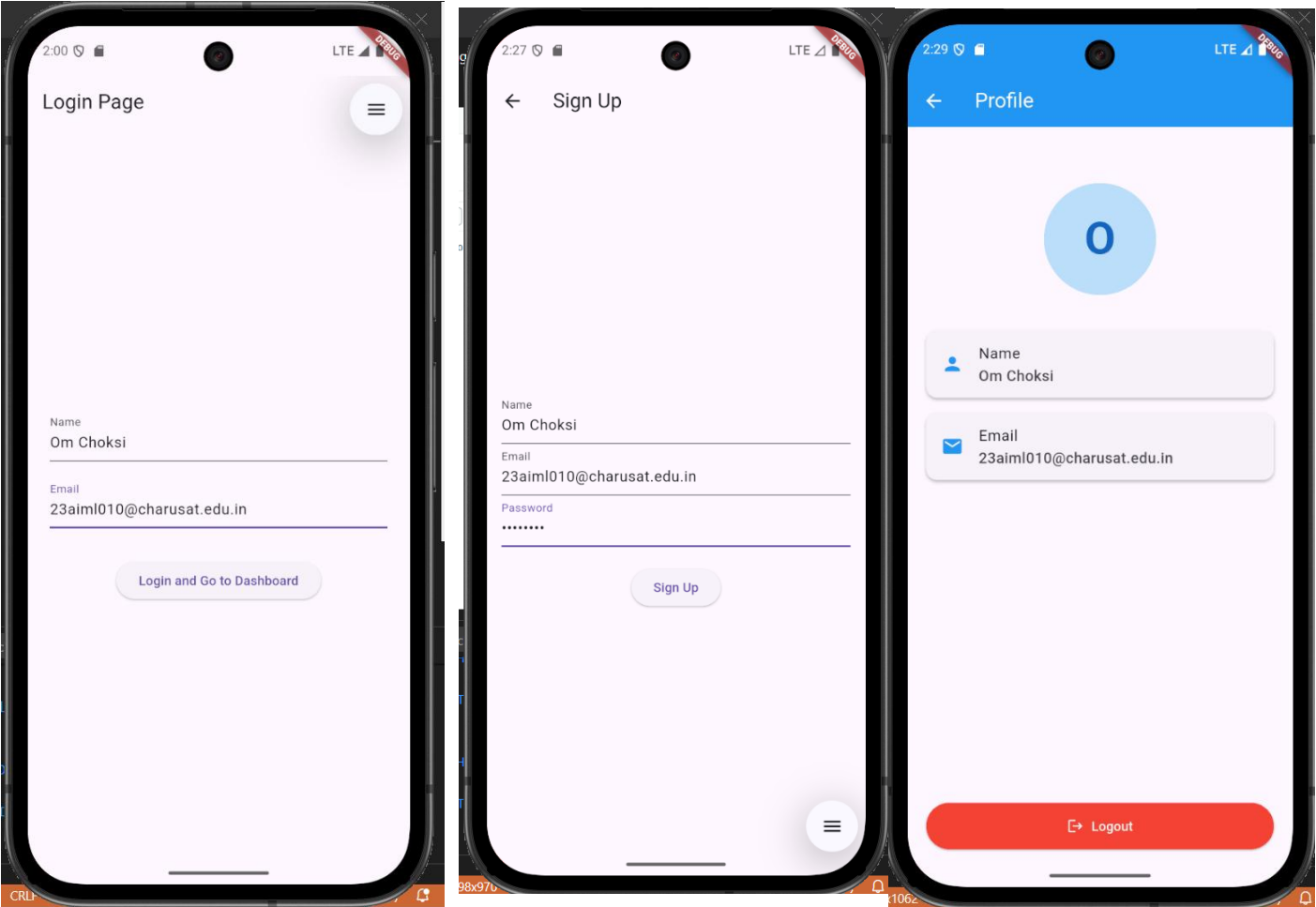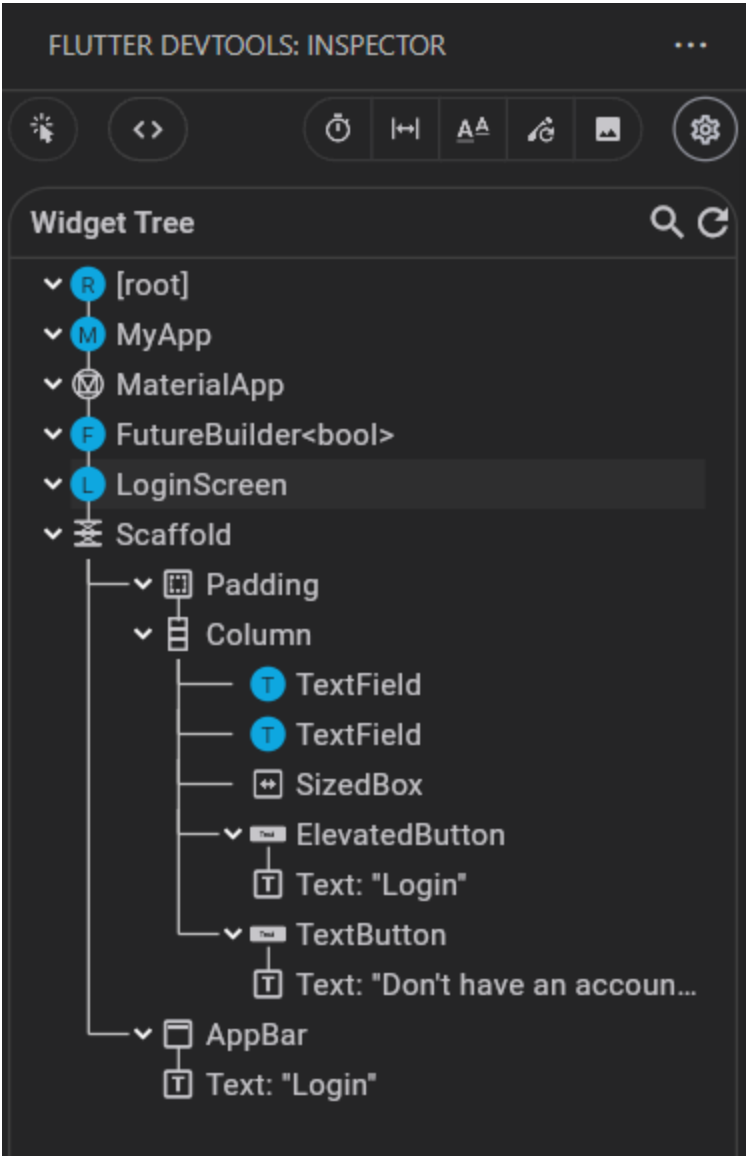
File Structure:

```
lib/
├────── main.dart
├────── home_screen.dart
├────── login_screen.dart
├────── profile_screen.dart
├────── services/
│       └────── auth_service.dart
└────── signup_screen.dart

backend/

├────── .env
├────── .gitignore
├────── check-db.js
├────── package.json
├────── package-lock.json
├────── server.js
├────── models/
│       └────── User.js
├────── routes/
│       └────── auth.js
```
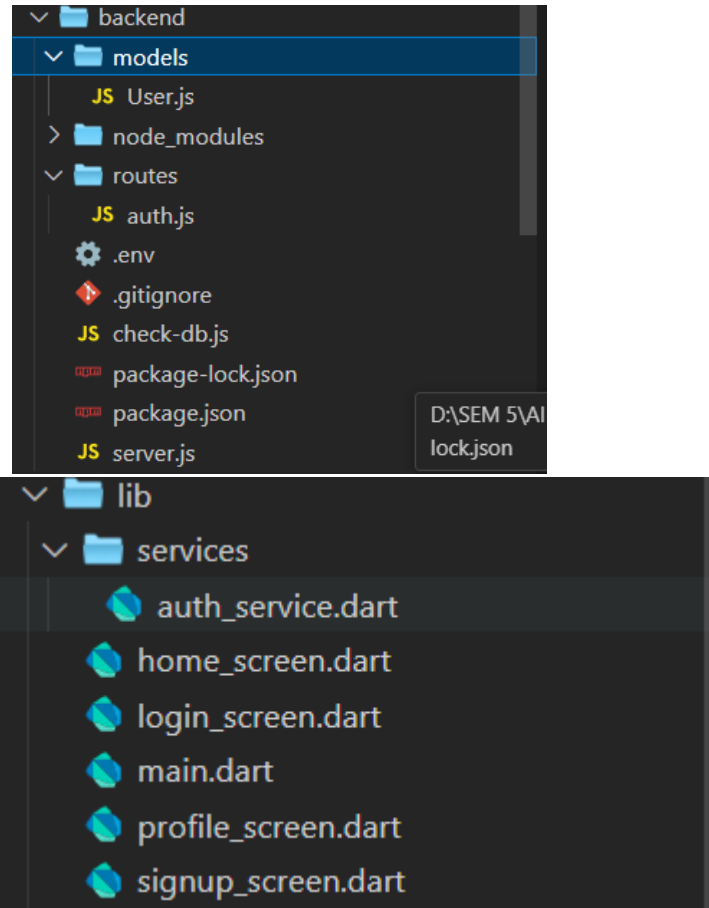
Screenshots:
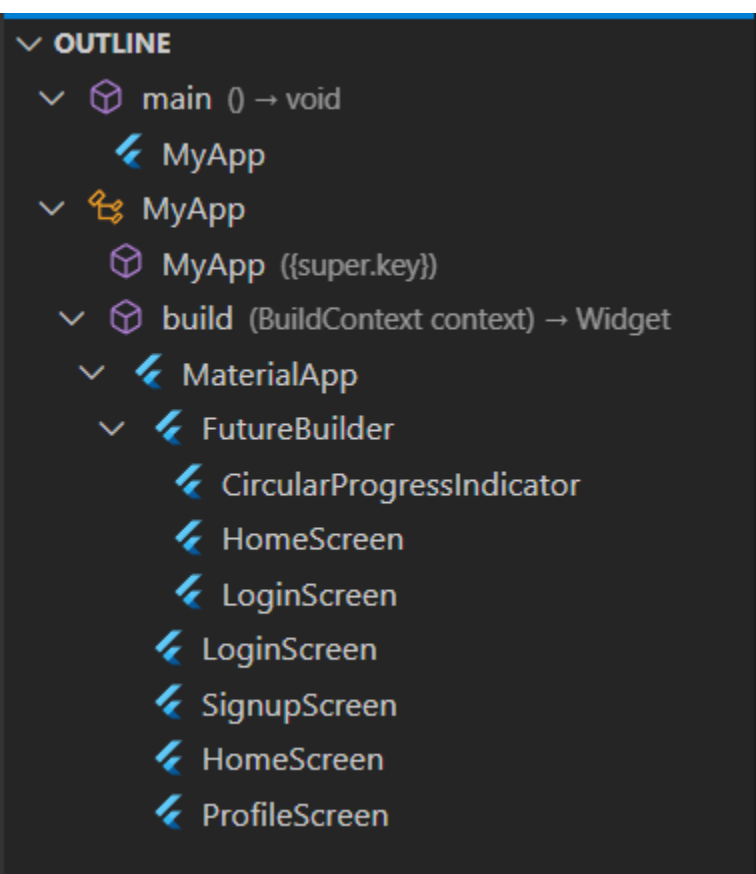
Key Questions:

1. How to send login credentials securely?
Ans: Credentials are sent via HTTPS POST requests with JSON payload containing email and password, using proper headers and secure token-based authentication with JWT for subsequent requests.

2. How to store token/session?
Ans: JWT tokens and user data are stored locally using SharedPreferences, with automatic session restoration on app restart and complete cleanup on logout.

3. How to handle invalid login?
Ans: Invalid login attempts return appropriate HTTP status codes (400) with error messages, which are parsed from JSON response and displayed to users via SnackBar notifications with user-friendly error text.

Key Skills to be Understand:

Auth Logic, API calls, Sessions

Implement login flow using APIs