SKILLER

1 Application Detail

Skiller is a Flutter-based mobile application that provides users with a list of available courses in various fields. It offers functionalities such as viewing course details, adding courses to favorites, enrolling in courses, and adding new courses. The app is integrated with Firebase for authentication and course data management, and uses SQLite to manage favorite courses locally. It also supports multi-language localization (English and Tamil) to cater to a diverse user base. The app features a clean, interactive UI/UX design aimed at delivering a professional user experience.

2 Workflow

2.1 Initial User Access

- **Step 1:** When the user first opens the app, the Startup Page appears, showcasing an interactive splash screen.
    - The startup screen is shown only the first time the user opens the app.
    - After this, it will not appear again, and the user is directed to the Login Page.

2.2 User Authentication (Login/Sign Up)

- **Step 2:** If the user is not logged in, they are prompted to either Log In or Register.
    - The Login and Register pages use Firebase Authentication for secure user sign-in and sign-up.
    - The user enters credentials (email, password) and submits.
    - On successful authentication, the user is redirected to the Home Page.
    - If authentication fails, an error message is displayed.

2.3 Home Page

- **Step 3:** After successful login, the user is presented with the Home Page.
    - An API call is made to Firebase to fetch the list of available courses.

- The homepage displays a list of courses with course image, title, and author.
- Users can click on any course to view its detailed page.

2.4 Course Detail Page

- **Step 4:** Upon selecting a course from the homepage, the user is redirected to the Detailed Course Page.
  - This page displays:
    - Course image, title, author, price, and what you'll learn.
  - Users can mark a course as a favorite by clicking the heart icon.
  - If the user is logged in and the course is not already in their favorites, they can also enroll by clicking the Enroll Button.

2.5 Favorites Page

- **Step 5:** The Favorites icon in the navigation bar shows the list of courses marked as favorites.
  - The list of favorite courses is stored in SQLite for local management.
  - Users can tap any favorite course to view its detailed page again.

2.6 Add Course Page

- **Step 6:** The Add Course icon allows logged-in users to add new courses.
  - Add Course form includes fields like course title, instructor, description, duration, image URL, and price.
  - Once the form is completed, the course is submitted to Firebase, and the new course is added to the homepage dynamically without a manual refresh (thanks to Provider state management).

2.7 User Logout

- **Step 7:** The User icon in the navigation bar provides an option to Log Out.
  - Upon logout, the user's session is cleared, and they are redirected to the Login Page.

2.8 Error Handling

- **Step 8:** If any error occurs (e.g., failed login, invalid course data submission), an error message is displayed.

- For example, "Login failed" or "Please fill out all required fields" messages will be shown to the user.

2.9 Real-Time Data Updates

- **Step 9:** After actions like adding a new course, changing language, or marking a course as a favorite, the app automatically updates the data.
  - This ensures the home page reflects real-time changes, such as new courses being added or courses being marked as favorites.

3 Flutter Concepts

3.1 Navigation

- Navigator is used to navigate between screens (e.g., from Home Page to Course Detail Page, or from Login Page to Register Page).
- Named Routes and pushNamed are used for efficient navigation between the app's primary screens.

3.2 Events

- Provider is used for state management to handle events like adding courses, changing favorites, and enrolling in courses.
- UI changes are made dynamically based on user actions such as tapping the favorite button or adding a new course.

3.3 Images

- Images are used throughout the app, such as in the Home Page and Course Detail Page.
- CachedNetworkImage is used for better performance when loading course images from URLs.
- Images are displayed alongside course details, including title, author, and what you will learn.

3.4 Layouts

- The app uses Flutter Layout Widgets (e.g., Column, Row, ListView, GridView) to structure the content on the Home Page, Course Detail Page, and Favorites Page.
- Responsive design ensures the app works smoothly on various screen sizes, from phones to tablets.

3.5 Database (Firebase)

- Firebase serves as the backend for managing user data and course data.
    - Firebase Authentication is used for secure login and registration.
    - Firebase Firestore stores and retrieves course data dynamically (e.g., course title, image, description).
    - The app uses Cloud Firestore queries to fetch and display courses in real-time.

3.6 Localization (Multi-Language Support)

- The app supports both English and Tamil through the Flutter localization system.
- intl package is used for text localization, ensuring that the app's main content changes dynamically based on the selected language.
- Language can be switched by the user, and the entire content (such as form field placeholders, buttons, and navigation elements) is updated in the selected language.
- **Note**: The course titles and descriptions remain unaffected by language changes since they are fetched from Firebase