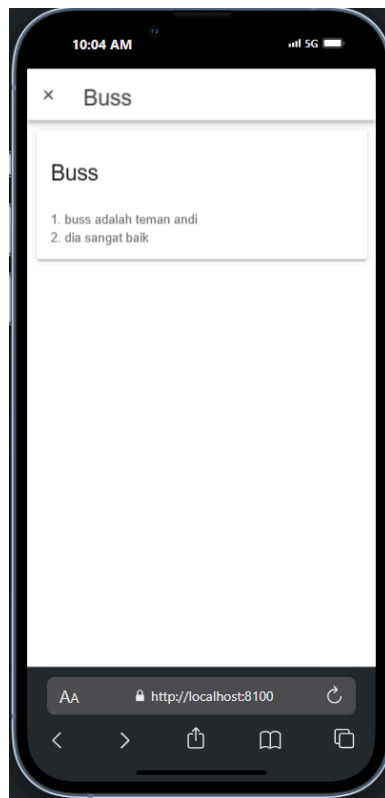
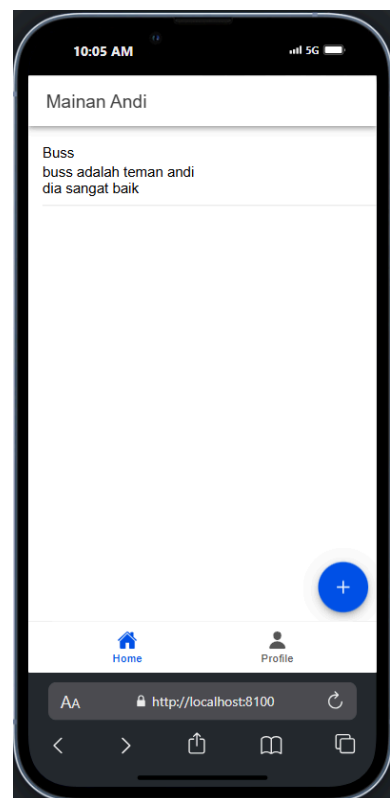


Halaman Profile



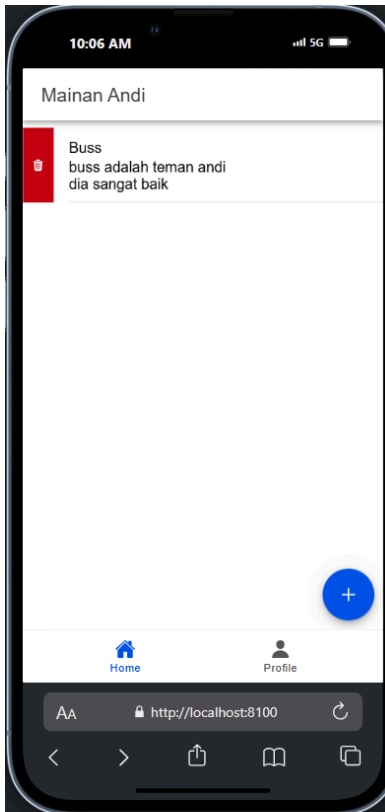
Halaman Detail



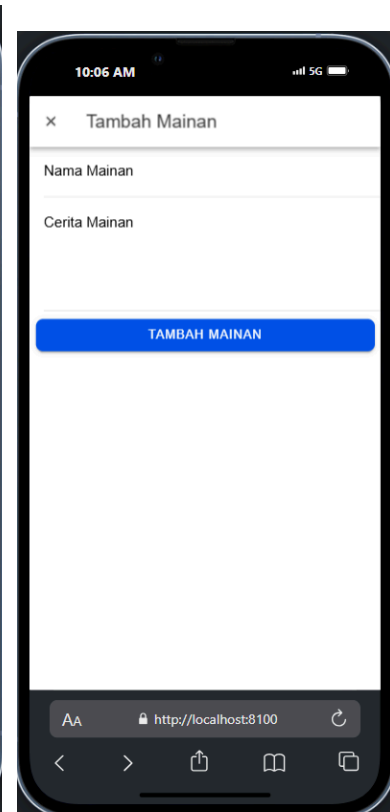
Halaman Home



Halaman Edit



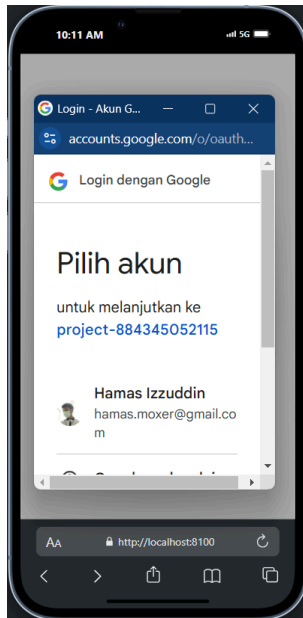
Halaman Hapus



Halaman Tambah

Nama : Hamas Izzuddin Fathi
NIM : H1D022097
Shift Baru : E
Shift KRS : A

1. Proses Login



- Saat pengguna mengklik tombol "Sign In with Google", fungsi loginWithGoogle dipanggil.
- Aplikasi menggunakan isPlatform('capacitor') untuk mendeteksi apakah aplikasi berjalan di platform mobile (iOS/Android) atau web.
- Aplikasi membuat instance GoogleAuthProvider untuk otentikasi Google di Firebase.
- Dengan signInWithPopup, Firebase memunculkan popup Google login, di mana pengguna dapat login menggunakan akun Google mereka.
- Jika berhasil, Firebase mengembalikan objek result yang berisi data pengguna yang berhasil login.
- Setelah proses login berhasil (baik pada platform web atau mobile), result.user berisi informasi akun Google pengguna yang sedang login, seperti displayName (nama pengguna), email, photoURL (foto profil), dan lainnya.
- Data pengguna ini disimpan dalam variabel user, yang merupakan ref dengan tipe
- User dari Firebase, sehingga dapat diakses di seluruh aplikasi melalui store auth.

2. Halaman Utama

- Setelah login berhasil, aplikasi akan menuju ke halaman utama
- dimana di halaman utama dapat menambahkan data mainan andi
- dan ada juga list mainan andi beserta ceritanya

- dan juga dapat menghapus data
- dan juga dapat mengedit data yang telah ada
- dan dibagian bawah terdapat Tabs Menu yang dimana ada Home dan Profile

3. Profile

- Setelah pengguna berhasil login, data pengguna dari akun Google disimpan dalam objek user.
- Objek ini berisi informasi akun, seperti displayName (nama pengguna) dan photoURL (URL foto profil), yang dapat diakses melalui user.value.displayName dan user.value.photoURL.
- Aplikasi dapat menggunakan informasi ini untuk menampilkan nama pengguna dan foto profil di berbagai halaman atau komponen.

4. Tambah Nama Mainan

- Ketika tombol Tambah pada Floating Action Button (FAB) ditekan, modal input (InputModal) akan terbuka. Modal ini menerima data mainan yang baru, dengan form input untuk nama mainan dan langkah-langkah pembuatan mainan.
- Pengguna mengisi form untuk nama mainan dan langkah-langkahnya. Ketika tombol submit diklik, data tersebut akan dikirimkan melalui event **@submit** ke fungsi **handleSubmit**. - Jika **editingId** adalah null (karena ini adalah mainan baru), fungsi **addMainan** dipanggil untuk menambahkan mainan ke Firestore dengan informasi dari form input.
- Setelah data berhasil disimpan, modal ditutup, dan daftar mainan diperbarui.

5. Masuk ke List Mainan

- Daftar mainan ditampilkan menggunakan komponen ion-list. Setiap mainan akan ditampilkan dengan nama dan langkah-langkah yang diformat (menggunakan **formattedStepsForList** untuk menampilkan langkah-langkah dengan pemisah baris).
- Ketika pengguna menekan item mainan, fungsi **goToDetail** dipanggil untuk membuka modal detail. Modal ini menunjukkan detail mainan yang lebih lengkap, termasuk nama mainan dan langkah-langkah yang diformat dengan nomor urut.
- Modal ini dibuka dengan menggunakan properti **isModalOpen** yang diubah menjadi true. Pengguna dapat melihat mainan secara lengkap di dalam modal tersebut.

6. Detail Mainan

- Modal detail ini digunakan untuk menampilkan informasi lengkap tentang mainan yang dipilih, termasuk nama mainan dan langkah-langkahnya yang diformat dengan nomor urut. Modal ini akan muncul di layar setelah **goToDetail** dipanggil.
- **currentMainan**: Variabel ini menyimpan data mainan yang dipilih (seperti nama dan langkah-langkah). Data ini digunakan untuk di-render dalam modal detail.

- **isModalOpen**: Variabel ini mengontrol apakah modal sedang ditampilkan atau tidak. Modal akan muncul jika nilainya true, dan hilang ketika nilainya false.
- Pengguna dapat menutup modal dengan menekan tombol "Tutup" yang ada di bagian header modal. Tombol ini akan memicu **isModalOpen = false**, yang akan menutup modal.

7. Update Mainan

- Jika pengguna ingin mengedit mainan, mereka menekan tombol edit yang ada pada opsi samping dari item mainan di dalam ion-item-options. Fungsi handleEdit dipanggil.
- Fungsi handleEdit akan memuat data mainan yang akan diedit, menyetel editingId dengan ID mainan yang dipilih, dan menampilkan modal input dengan data mainan yang sudah ada.
- Modal input akan menampilkan data yang sudah ada untuk mainan yang sedang diedit, memungkinkan pengguna untuk mengubahnya.
- Setelah pengguna mengubah data dan menekan tombol submit, data baru akan disimpan dengan memanggil fungsi updatemainan untuk memperbarui mainan yang ada di Firestore. Daftar mainan akan diperbarui.

8. Delete Mainan

- Pengguna dapat menghapus mainan dengan menekan tombol hapus pada opsi samping item mainan di dalam ion-item-options.
- Fungsi handleDelete akan dipanggil, yang menghapus mainan dari Firestore dengan menggunakan deletemainan.
- Setelah mainan dihapus, daftar mainan diperbarui dengan memanggil loadmainans untuk memuat data mainan terbaru

SOAL

1. Jelaskan alasan mengapa Ionic framework tidak sepopuler framework mobile lainnya!
 - a. Performa yang Kurang Optimal
Ionic menggunakan webview untuk merender aplikasi, yang berarti aplikasi Ionic pada dasarnya adalah aplikasi web yang dijalankan dalam sebuah kontainer.
 - b. Dependensi pada WebView
Karena Ionic menggunakan WebView untuk merender tampilan aplikasi, ada ketergantungan pada performa WebView dan browser di perangkat.
 - c. Kurangnya Dukungan Penuh untuk Fitur Native
Meskipun Ionic menyediakan banyak plugin untuk mengakses fitur perangkat, namun kadang-kadang Ionic tidak dapat sepenuhnya menggantikan pengalaman aplikasi native seperti React Native atau

Flutter, yang lebih baik dalam mengakses fitur perangkat keras secara langsung.

2. Sebutkan fungsi dari komponen ion refresher dalam Ionic ?
 - a. **Menyegarkan Konten:** Pengguna dapat menarik layar ke bawah untuk memicu penyegaran konten, biasanya digunakan untuk mengambil data terbaru dari server.
 - b. **Event Pengendalian:** ion refresher memungkinkan pengembang untuk mendengarkan event seperti `ionRefresh`, yang akan dipicu saat pengguna menarik layar ke bawah, dan menyediakan callback untuk menyelesaikan proses penyegaran (misalnya, setelah mengambil data baru)
3. Bagaimana cara mengatur agar pengguna yang sudah login tidak bisa kembali ke halaman login dan pengguna yang belum login tidak bisa masuk ke dalam halaman home atau dashboard?
 - a. Menggunakan Vue Router dan Local Storage (Contoh untuk Vue)
 - **Setelah Pengguna Login:** Simpan status login (misalnya, token atau ID pengguna) ke dalam `localStorage` atau `sessionStorage`.
 - **Cek Status Login di Router:** Gunakan **navigation guards** pada router untuk memeriksa apakah pengguna sudah login sebelum mengakses halaman tertentu, dan arahkan mereka ke halaman login jika mereka belum login.
 - **Logout Pengguna:** Ketika pengguna logout, hapus status login dari `localStorage`: