JAM.26

DAY 2 FULL-STACK CHECKLIST

Day 2 - Full-Stack Development & Debugging

Authentication, persistent data, user ownership, security, and evidence-based debugging.

01 Before You Start

Day 1 Project > Local Test > Git Status > Firebase Account

[ ]	Day 1 project opens and runs locally
[ ]	Core frontend feature still works
[ ]	Visual Studio Code and terminal are ready
[ ]	Node.js and npm are working
[ ]	Git repository is clean or current changes are understood
[ ]	Latest working checkpoint is committed
[ ]	Firebase / Google account is available
[ ]	Browser Developer Tools can be opened

CHECKPOINT:

Your Day 1 application is stable enough to connect to backend services.

02 Firebase Authentication

Firebase Setup > Email/Password > Sign Up > Sign In > Sign Out

[ ]	Firebase web application created or selected
[ ]	Firebase SDK installed and initialized
[ ]	Email/Password Authentication enabled
[ ]	User can create an account
[ ]	User can sign in
[ ]	User can sign out
[ ]	Application detects authentication state
[ ]	Authentication errors are shown clearly
[ ]	Interface responds appropriately when no user is signed in

CHECKPOINT:

The application can reliably identify the current authenticated user.

03 Persistent Data with Firestore

User Action > Application Logic > Firestore > Stored Data > Reload

- [ ] Cloud Firestore database created
- [ ] Collection and document structure is understood
- [ ] Application can create a record
- [ ] Application can read stored records
- [ ] Application can update or delete records when required
- [ ] Stored data survives a page refresh
- [ ] Firestore paths and field names are consistent
- [ ] Application is not relying only on temporary local state

CHECKPOINT:

Application data persists beyond the current browser session.

04 User Ownership & Security

Authenticated User > UID > ownerId > Query + Security Rules > Authorized Data

- [ ] User-owned records store the authenticated user's UID as ownerId
- [ ] Application requests only the correct user's records
- [ ] Firebase Security Rules enforce access to user-owned records
- [ ] Authentication is not treated as database authorization by itself
- [ ] Frontend filtering is not treated as a security boundary
- [ ] Application tested with two different accounts
- [ ] One user cannot access another user's private records
- [ ] Permission errors are investigated instead of bypassed with public rules

CHECKPOINT:

Identity, data ownership, and database authorization work together correctly.

05 Full-Stack Debugging

Expected > Observed > Evidence > Hypothesis > Fix > Test > Verify

[ ]	Expected behavior is written clearly
[ ]	Actual behavior is identified precisely
[ ]	Browser console checked for errors
[ ]	Network panel checked when requests are involved
[ ]	Terminal output checked
[ ]	Authentication state and current UID checked
[ ]	Firestore path, query, and stored data checked
[ ]	Security Rules checked when requests are denied
[ ]	First failing layer is identified before changing code
[ ]	AI receives real evidence instead of only a symptom
[ ]	Proposed fix is tested before being trusted
[ ]	Working result is verified after the fix

CHECKPOINT:

The bug is diagnosed from evidence instead of random code changes.

06 AI Workflow & Optional Hooks

Inspect > Direct AI > Review > Run > Test > Debug > Verify > Automate Carefully

[ ]	AI inspects the existing project before major changes
[ ]	Requests describe one clear objective
[ ]	Generated changes are reviewed
[ ]	Application is run after changes
[ ]	Feature behavior is tested manually
[ ]	Errors are investigated with evidence
[ ]	Working changes are committed to GitHub
[ ]	Repetitive checks are identified only after the workflow is understood
[ ]	Hooks are used only when they improve testing, quality, or productivity
[ ]	Automations are understood before they are enabled

CHECKPOINT:

AI accelerates the workflow without replacing review, testing, or engineering judgment.

FINAL FULL-STACK PATH

LOCAL APP > AUTHENTICATION > FIRESTORE > USER OWNERSHIP

SECURITY RULES > DEBUG > TEST > VERIFY > COMMIT

DAY 2 COMPLETE

A user can authenticate, application data persists, private records are protected by ownership and Security Rules, and full-stack failures can be investigated using evidence.