### **Test Strategy**

* **Testing Methodology**:  
   Manual testing with console logging, Supabase DB inspection, and browser inspection.  
   Server console logs used for backend debugging.  
   Supabase dashboard used to confirm correct data sync and reflection.
* **Unit Testing Focus Areas**:  
  + Inventory management (add, edit, remove)
  + Notification delivery
  + Household sharing and control
  + Profile/Setting creation, editing, and saving
  + Recipe suggestion engine
  + Inventory sorting and filtering
  + Household invitations via to other accounts
  + Selecting notification preferences
* **Environments Tested**:  
  + **Frontend**: React app in browser (Chrome, Firefox)
  + **Backend**: Supabase dashboard, API logs
  + **Notification**: Supabase Auth/Triggers and email delivery
* **Tools Used**:  
  + Browser Developer Console
  + Supabase Dashboard
  + React DevTools
  + Manual trial & error debugging with console.log() and server console output

**Test Report**

### 

| **Feature** | **Test Description** | **Test Steps** | **Expected Result** | **Pass/Fail** |
| --- | --- | --- | --- | --- |
| Add Inventory Item | Add a food item with details | Add item via UI → Check browser → Confirm in Supabase | Item appears in UI and Supabase | Pass |
| Edit Inventory Item | Modify details of item and save | Edit item via UI → Confirm updated data in browser and Supabase | Edited item displays updated details | Pass |
| Remove Inventory Item | Delete item from inventory | Click delete → Confirm prompt → Check UI and Supabase | Item no longer exists in UI or Supabase | Pass |
| Email Notification | Check expiration notification is sent | Add item expiring soon → Wait → Check inbox | Email alert received | Pass |
| Household Inventory Sharing | Share inventory with another user | Invite user → User logs in → Check shared access | Shared inventory visible to second user | Pass |
| Household Edit Control | Verify shared users can edit inventory | Logged in as second user → Edit item | Changes appear for both users | Pass |
| Recipe Suggestions | Suggest recipes from current inventory | Add items → Open recipe section | Relevant recipes suggested | Pass |
| Item Filtering | Sort by name or other item detail | Apply filters in UI | Filtered items shown accurately | Pass |
| Profile Creation | Create a user profile with login | Register new user via Supabase Auth | Profile successfully created | Pass |
| Edit Profile Details | Update profile name | Edit profile settings via UI | Changes saved and shown correctly | Pass |
| Edit Household Details | Change household name | Open household settings → Edit value | Household details updated | Pass |
| Send User Invitations | Invite others via email and have it appear in their account | Enter email → Submit invite → Check if invited user appears | Invitation sent, user added | Pass |
| Notification Preferences | Set preferences for expiration/addition alerts | Toggle notification types in setting | Correct notifications triggered based on settings | Pass |

**Notes**

* Debugging was iterative with heavy use of console.log() and trial/error-based tracing.
* Supabase integration worked reliably once endpoints and listeners were corrected.
* The recipe engine performed using basic filtering logic.
* Email notifications were verified via inbox receipts from test accounts.