BrewPOS - Complete Development Plan

📋 Overview

A comprehensive guide to systematically build a complete POS system with Firebase integration and offline support.

Phase 1: Core Architecture & Navigation (MILESTONE 1)

Issue #1: Setup React Router and Navigation Context

Priority: 🔴 HIGH
Description: Implement React Router v6 with route management and create a navigation context for state management.
Tasks:

- □ Install react-router-dom v6
- □ Create routing structure (Dashboard, Orders, Products, Reports, Settings)
- □ Create NavContext for active screen state
- □ Setup route parameters for sub-routes **Related Files:**
- `src/App.tsx` - Main routing
- `src/context/NavContext.tsx` - Navigation state
- `src/pages/*.tsx` - Page components **Acceptance Criteria:**
- Routing works without full page reloads
- Context properly tracks active screen
- Browser back/forward buttons work

Issue #2: Fix Sidebar Component with Active States

Priority: 🔴 HIGH
Description: Implement a fully functional sidebar with active link highlighting, collapsible menus, and proper spacing.
Tasks:

- □ Create proper sidebar data structure with nested items
- □ Implement active state tracking using useLocation hook
- □ Fix gap between top and bottom navigation sections
- □ Add smooth transitions and hover effects
- □ Ensure mobile responsiveness with menu toggle **Related Files:**
- `src/components/app-sidebar.tsx`
- `src/components/ui/sidebar.tsx` **Acceptance Criteria:**
- Active links are highlighted
- Hover effects work smoothly
- Menu collapses properly on mobile
- Spacing is consistent and visual

Issue #3: Create Page Component Structure

Priority: 🔴 HIGH
Description: Create all main page components that sidebar will render. **Tasks:**

- □ Dashboard page with statistics
- □ Orders/Main Order page

- ☐ Products/Manage Dishes page
- ☐ Analytics/Reports page
- ☐ Withdrawals page
- ☐ Manage Payments page
- ☐ Settings page **Related Files:**
- `src/pages/Dashboard.tsx`
- `src/pages/Orders.tsx`
- `src/pages/Products.tsx`
- `src/pages/Analytics.tsx`
- `src/pages/Withdrawals.tsx`
- `src/pages/Payments.tsx`
- `src/pages/Settings.tsx` **Acceptance Criteria:**
- All pages render without errors
- Each page is accessible from sidebar
- Page content updates when clicking sidebar items

Phase 2: Product Management (MILESTONE 2)

Issue #4: Create Product Management UI

Priority: 🔴 HIGH
Description: Build the product management/menu management interface. **Tasks:**

- ☐ Create product list view with search and filter
- ☐ Implement add product modal form
- ☐ Create edit product functionality
- ☐ Add delete product with confirmation
- ☐ Implement category management
- ☐ Add product image upload **Related Files:**
- `src/pages/Products.tsx`
- `src/components/pos/ProductForm.tsx`
- `src/components/pos/ProductList.tsx` **Acceptance Criteria:**
- CRUD operations work for products
- Image uploads work
- Validation on form fields
- Success/error messages display

Issue #5: Setup Local Database (IndexedDB)

Priority: 🟡 MEDIUM
Description: Configure IndexedDB for offline product storage. **Tasks:**

- ☐ Initialize IndexedDB with proper schemas
- ☐ Create helper functions (add, get, update, delete, getAll)
- ☐ Setup product store with proper indexing
- ☐ Add migration support for future schema changes **Related Files:**
- `src/lib/idb.ts`
- `src/hooks/useLocalDB.ts` **Acceptance Criteria:**
- Data persists after page reload
- Queries return correct data
- No errors in console

Phase 3: Order Management (MILESTONE 3)

Issue #6: Enhance POS/Order Screen

Priority: 🔴 HIGH
Description: Improve the main order/POS screen with better UX. **Tasks:**

- □ Add real-time order totals with tax calculation
- □ Implement quantity modifiers (keyboard shortcuts, +/- buttons)
- □ Add customer selection/notes field
- □ Implement order discounts/coupons
- □ Add table selection for dine-in orders
- □ Implement payment method selection **Related Files:**
- `src/pages/POSScreen.tsx`
- `src/components/pos/OrderSidebar.tsx` **Acceptance Criteria:**
- Order calculations are accurate
- UI is fast and responsive
- All interactions feel smooth

Issue #7: Order History and Management

Priority: 🟡 MEDIUM
Description: Create order history, viewing, and reordering functionality. **Tasks:**

- □ Display order history with filters
- □ Show order details (items, time, payment, status)
- □ Implement reorder from history
- □ Add receipt printing
- □ Create order search functionality **Related Files:**
- `src/pages/Orders.tsx`
- `src/components/pos/OrderHistory.tsx` **Acceptance Criteria:**
- Orders display with all details
- Reorder functionality works
- Search and filters work

Phase 4: Analytics & Reports (MILESTONE 4)

Issue #8: Analytics Dashboard

Priority: 🟡 MEDIUM
Description: Create comprehensive analytics and reporting dashboard. **Tasks:**

- □ Sales summary widget
- □ Revenue trends chart
- □ Top products widget
- □ Hourly/daily/weekly/monthly reports
- □ Category-wise sales breakdown
- □ Payment method analysis **Related Files:**
- `src/pages/Analytics.tsx`
- `src/components/dashboard/Charts.tsx` **Acceptance Criteria:**
- Charts display correctly
- Filters work properly
- Data is accurate

Issue #9: Export Reports

Priority: 🟢 LOW
Description: Add export functionality for reports. **Tasks:**

- □ Export to CSV
- □ Export to PDF with formatting
- □ Email report functionality
- □ Scheduled report generation **Related Files:**
- `src/utils/exportReports.ts` **Acceptance Criteria:**
- Files generate without errors
- Data is complete and formatted

Phase 5: Firebase Integration (MILESTONE 5)

Issue #10: Setup Firebase Project

Priority: 🔴 HIGH
Description: Configure Firebase project and authentication. **Tasks:**

- □ Create Firebase project
- □ Setup Firebase configuration
- □ Enable Authentication (Email/Password, Google)
- □ Setup Firestore database
- □ Configure Storage bucket
- □ Setup Security Rules **Related Files:**
- `src/lib/firebase.ts`
- `.env.local` - Environment variables **Acceptance Criteria:**
- Firebase initializes without errors
- Credentials are secure

Issue #11: Implement Firebase Authentication

Priority: 🔴 HIGH
Description: Integrate Firebase auth with login/logout functionality. **Tasks:**

- □ Replace mock auth with Firebase auth
- □ Implement login with email/password
- □ Add Google login option
- □ Implement logout
- □ Add session persistence
- □ Create protected routes
- □ Add password reset functionality **Related Files:**
- `src/hooks/useAuth.ts`
- `src/pages/Login.tsx` **Acceptance Criteria:**
- Login/logout works
- Session persists on refresh
- Unauthenticated users can't access POS

Issue #12: Sync Orders to Firestore

Priority: 🔴 HIGH
Description: Setup real-time Firestore sync for orders. **Tasks:**

- □ Create Firestore collection schema for orders
- □ Implement order save to Firestore
- □ Add real-time listener for orders
- □ Implement batch sync functionality
- □ Add conflict resolution **Related Files:**
- `src/lib/firebase.ts`
- `src/hooks/useFirestore.ts` **Acceptance Criteria:**
- Orders save to Firestore
- Real-time updates work
- Conflicts resolve properly

Issue #13: Sync Products to Firestore

Priority: 🟡 MEDIUM
Description: Setup product synchronization with Firestore. **Tasks:**

- □ Create products collection in Firestore
- □ Sync local products to cloud
- □ Implement product updates from cloud
- □ Add real-time product listener
- □ Handle image uploads to Storage **Related Files:**
- `src/lib/firebase.ts`
- `src/hooks/useFirestore.ts` **Acceptance Criteria:**
- Products sync properly
- Images upload and display
- Changes propagate in real-time

Phase 6: Offline Functionality (MILESTONE 6)

Issue #14: Implement Offline Detection

Priority: 🟡 MEDIUM
Description: Add network status detection and UI indicators. **Tasks:**

- □ Detect online/offline status
- □ Show sync indicator in sidebar
- □ Implement offline banner
- □ Queue operations when offline **Related Files:**
- `src/hooks/useOnlineStatus.ts`
- `src/components/OfflineBanner.tsx` **Acceptance Criteria:**
- Status updates correctly
- UI clearly indicates offline state

Issue #15: Offline Order Queuing

Priority: 🟡 MEDIUM
Description: Queue and retry orders when offline. **Tasks:**

- □ Save orders to IndexedDB when offline
- □ Queue failed Firestore writes
- □ Implement sync on reconnect
- □ Retry failed transactions
- □ Show pending sync count **Related Files:**
- `src/hooks/useOfflineSync.ts` **Acceptance Criteria:**
- Orders are saved offline
- Sync works when back online
- No data loss

Issue #16: Offline Analytics

Priority: 🟢 LOW
Description: Calculate analytics from offline data. **Tasks:**

- □ Load cached data for charts
- □ Show "last updated" timestamp
- □ Merge online and offline data
- □ Warn about incomplete data **Related Files:**
- `src/hooks/useAnalytics.ts` **Acceptance Criteria:**
- Analytics work offline
- Data accuracy indicated

Phase 7: Payments & Transactions (MILESTONE 7)

Issue #17: Integrate Payment Gateway

Priority: 🟡 MEDIUM
Description: Setup payment processing (Toyyibpay for Malaysia). **Tasks:**

- □ Setup Toyyibpay integration
- □ Implement payment modal
- □ Handle payment callbacks
- □ Store payment records
- □ Add receipt generation **Related Files:**
- `src/lib/toyyibpay.ts`
- `src/components/PaymentModal.tsx` **Acceptance Criteria:**
- Payments process correctly
- Receipts generate properly
- Payment history tracked

Issue #18: Manage Payment Methods

Priority: 🟢 LOW
Description: Support multiple payment methods. **Tasks:**

- □ Cash payments
- □ Card payments
- □ Digital wallet support
- □ Split payments
- □ Payment reconciliation **Related Files:**
- `src/pages/Payments.tsx` **Acceptance Criteria:**

- All payment methods work
- Reconciliation is accurate

Phase 8: Business Operations (MILESTONE 8)

Issue #19: Withdrawals Management

Priority: 🟢 LOW
Description: Track and manage staff withdrawals. **Tasks:**

- □ Create withdrawal tracking system
- □ Implement approval workflow
- □ Track by user and date
- □ Generate withdrawal reports **Related Files:**
- `src/pages/Withdrawals.tsx` **Acceptance Criteria:**
- Withdrawals tracked correctly
- Reports are accurate

Issue #20: Settings & Configuration

Priority: 🟢 LOW
Description: Create comprehensive settings page. **Tasks:**

- □ Restaurant info settings
- □ Tax configuration
- □ Receipt customization
- □ User management
- □ Backup/Restore functionality
- □ Notification preferences **Related Files:**
- `src/pages/Settings.tsx` **Acceptance Criteria:**
- All settings save properly
- Changes apply immediately

Phase 9: Performance & Polish (MILESTONE 9)

Issue #21: Performance Optimization

Priority: 🟡 MEDIUM
Description: Optimize app performance and loading times. **Tasks:**

- □ Code splitting for routes
- □ Image optimization
- □ Lazy loading components
- □ Memoization where needed
- □ IndexedDB indexing optimization **Related Files:**
- All components **Acceptance Criteria:**
- Lighthouse score > 90
- Page load < 2s on 3G

Issue #22: Bug Fixes & Testing

Priority: 🔴 HIGH
Description: QA testing and bug fixes. **Tasks:**

- □ Manual testing on multiple devices
- □ Browser compatibility testing
- □ Offline mode stress testing
- □ Edge case handling
- □ Error message improvements **Acceptance Criteria:**
- No critical bugs
- App stable on all devices

Recommended Sidebar Components

```
BrewPOS Sidebar Structure:
├── Header (Logo + Version)
├── Search Bar
├── Navigation Menu
│   ├── Dashboard (with icon)
│   ├── Main Order (collapsible)
│   │   ├── Main Menu
│   │   ├── Beverages
│   │   ├── Desserts
│   │   └── Appetizers
│   ├── Analytics (with icon)
│   ├── Withdrawals (with icon)
│   ├── Manage Dishes (collapsible)
│   │   ├── Add New Dish
│   │   ├── Edit Existing
│   │   └── Remove Dish
│   └── Manage Payments (with icon)
├── Divider
└── Footer
    ├── User Info
    ├── Online Status Indicator
    └── Logout Button
```

Implementation Priority Order

1. **Phase 1** - Complete first (foundation)
2. **Phase 2** - Parallel with Phase 3
3. **Phase 3** - Core business logic

4. **Phase 5** - Backend services
5. **Phase 4** - Reporting
6. **Phase 6** - Offline support
7. **Phases 7-8** - Business features
8. **Phase 9** - Final polish

Success Metrics

✅ All issues completed
✅ App functions offline
✅ Real-time sync works
✅ All CRUD operations functional
✅ Performance score > 90
✅ Zero critical bugs
✅ Mobile responsive
✅ User can complete full POS workflow

Technology Stack

- **Frontend:** React 18 + TypeScript
- **Routing:** React Router v6
- **State:** Context API + Local State
- **Styling:** Tailwind CSS
- **Database:** Firebase Firestore + IndexedDB
- **Auth:** Firebase Auth
- **Payments:** Toyyibpay
- **UI Components:** shadcn/ui
- **Icons:** Lucide React