Basic Rental System Requirements

1. Functional Requirements

#1.1 Rental Management

- Users (admin/staff) can create a new rental for an available item, linking it to a customer and specifying start and expected return dates.

- Users can view a list of all rentals, with status (active, returned, overdue).

- Users can return an item, changing its status appropriately.

- Prevent rental of items already on rent (no double-booking).

- Users can filter or search rentals by status, customer, or item.

# 1.2 Customer Management

- Users can add, edit, view, and delete basic customer records (name, contact).

# 1.3 Inventory Integration

- Changing an item’s rental status updates its availability in the inventory.

- Rental status must be visible in the inventory list.

# 1.4 Data Validations

- Ensure all required fields are filled for rentals and customers.

- Prevent invalid or conflicting rental periods.

2. Technical Requirements

#2.1 Database Schema

\*\*Tables:\*\*

- \*\*rentals\*\*: rental\_id (PK), item\_id (FK), customer\_id (FK), start\_date, expected\_return\_date, actual\_return\_date, status

- \*\*customers\*\*: customer\_id (PK), name, contact

- \*\*items (updated)\*\*: status field added if not present

# 2.2 Backend Logic

- Rental creation, return, and listing endpoints in PHP

- Status updates for items on rent/return

- Customer CRUD operations

- Validations to prevent double-booking and ensure required fields

#2.3 Frontend Logic

- Forms for creating/returning rentals and managing customers

- Rental listing with filters/search

- Clear error and confirmation messages

# 2.4 Access Control

- Only admin/staff users manage rentals/customers

#2.5 Non-Functional Requirements

- Secure code (prepared statements, input validation)

- Responsive, simple UI (no frameworks required)

- Inline code comments and documentation

# 2.6 Out of Scope

- Payments, late fees, notifications, multi-item rentals, advanced analytics

---

#3. Acceptance Criteria

- Rentals can be created, viewed, and returned via the UI

- Double-booking is prevented

- Item and rental statuses update correctly

- Customers can be managed via the UI

- All actions provide user feedback