Master Rental App

Project Documentation Index

1. Project Introduction

The **Master RentApp** is a centralized rental management system designed for property administrators managing multiple owners and companies. It automates the rental lifecycle, from onboarding tenants to generating tax-compliant invoices and tracking payments.

---

2. Folder and File Structure

1. Root Directory

The root directory (**in sunnsun cPanel > snssolutions.in >RENTAL > master_rentapp**) contains the main page controllers that users visit in their browser.

- **index.php**: The main dashboard entry point. It checks user sessions and loads either the admin portal or the general dashboard.
- **Module Pages**: Files like billing.php, owner.php, tenant.php, and payment.php serve as the main interfaces for their respective modules.
- **Reporting Pages**: Files such as report.php, allreport.php, due-list.php, and pending-list.php generate various financial and status reports.
- **Document Generation**: invoice.php and receipt.php are likely used to generate or print invoices and payment receipts.
- **Authentication**: login.php handles user entry, while logout.php clears the session.

2. config/ (Configuration)

Contains core setup files for the application environment.

- **Database.php**: The most critical file. It manages the PDO database connection, session handling, and global utility functions like sendSMS, numtowords, and secure_text.

3. models/ (Business Logic)

These classes handle all database interactions (CRUD operations) for specific entities.

- **Billing.php**: Manages invoice generation, calculating taxes (CGST/SGST), and tracking bill statuses.
- **Login.php**: Handles user authentication, profile updates, and retrieving company-specific information.
- **Owner.php**: Manages data for property owners, including bank details and contact info.
- **Payment.php**: Handles recording of payments, TDS deductions, and updating payment statuses.
- **Report.php**: Contains complex queries for dashboard statistics, transaction summaries, and pending lists.
- **Tenant.php**: Manages tenant profiles, agreement dates, and calculates outstanding balances via database views.

4. partials/ (UI Components)

Contains reusable HTML/PHP snippets to maintain a consistent design across the app.

- **Layout**: header.php, footer.php, and navbar.php provide the common structure for every page.
- **Views**: Files ending in _view.php (e.g., billing_view.php, tenant_view.php) contain the specific HTML content and forms for their corresponding root pages.

5. Api/ (Backend Endpoints)

Contains subdirectories for each module (e.g., tenant/, billing/, payment/). Each subdirectory typically contains:

- **fetch.php / fetch_single.php**: Returns data in JSON format for the front-end.
- **create.php / update.php**: Processes form submissions to add or modify database records.
- **delete.php**: Handles record deletion (often via a "status" flag rather than a hard delete).

6. assets/ (Static Assets)

- **css/**: Contains admin.css and app-style.css for application styling.
- **js/**: Contains jQuery and custom scripts (e.g., billing.js, tenant.js, payment.js) that handle AJAX calls to the API and dynamic UI updates.
- **img/**: Stores icons, background images, and the application loader.

---

3. User Authentication (How to Login)

Security is handled through a structured login process defined in the API and models:

- **Verification Process**: The Api/login.php script receives a user_name and user_password. It calls Login::auth() to find the user in the database and uses password_verify() to check the encrypted password.
- **Session Management**: Once verified, details such as user_id, co_id (Company ID), and is_admin are saved in a "Session" to maintain the user's state. For example, $_SESSION['co_id'] ensures a user only sees data for their specific company.
- **Access Control**: session_private() ensures only logged-in users can see protected pages, while session_public() prevents logged-in users from seeing the login page again.

---

4. Tenant Management (Adding Renters)

The Tenant.php model acts as the central hub for all renter data, ensuring every tenant's profile is complete:

- **Detailed Onboarding**: Captures comprehensive data including **GSTIN** for tax invoices, and **Aadhar**, **PAN**, and contact info for legal identification. It also maps each tenant to a specific **Owner**, **Location**, and **Shop Number**.
- **Agreement Monitoring**: Tracks agreement periods and calculates **Expiry Date** to notify admins before a lease ends. The function can identify tenants whose leases expire within a specified window (e.g., next 30 days).
- **Financial Profile Setup**: Establishes the base **Rent Amount**, any **Opening Balance** and the **Security Deposit Amount**. It also includes a **TDS toggle** to flag if a tenant deducts tax at the source.

5. Billing & Invoicing

The Billing.php model generates professional, tax-compliant invoices:

- **Smart Invoicing**: The invoice_no() function generates sequential IDs following the April-to-March financial year cycle (e.g., Inv-01/2025-26).
- **Tax Compliance**: Automatically calculates and applies **CGST** and **SGST** amounts based on the base rent amount.
- **Duplicate Prevention**: The exist_bill() function checks if a bill already exists for a specific tenant and month to prevent double-billing.

---

6. Payment & Receipts (Money Collection)

The Payment.php model tracks income revenue:

- **Transaction Tracking**: Records the payment mode (Cash, Cheque, Bank), cheque numbers, bank names, and receipt numbers.
- **TDS Management**: Handles **Tax Deducted at Source (TDS)** separately, ensuring the total amount the tenant "paid" (Received + TDS) is correctly credited toward their balance.
- **Status Management**: Allows administrators to mark a payment as "Received," which updates the status in the system.

---

7. Reporting & Analytics (Visualizing Data)

The Report.php model summarizes complex financial information for administrators:

- **Tenant Transaction**: Provides a chronological "ledger view" for each tenant, combining all their bills and receipts into a single statement.
- **Dashboard Visuals**: The main dashboard fetches data to show total payments vs. total bills across different companies using pie charts.
- **Due & Pending Lists**: Summarizes total outstanding balances for all tenants to identify late payers and monitor company-wide debt.

---

8. Formulas & Logic (Plain Text)

The system uses automated calculations to ensure financial accuracy:

- **Step 1: Calculate GST Amounts**
  - Formula: (Rent Amount * GST Percentage) / 100
  - Example: (20,000 * 9) / 100 = **1,800** (for CGST and SGST individually)
- **Step 2: Calculate Sub-Total**
  - Formula: Rent Amount + CGST Amount + SGST Amount
  - Example: 20,000 + 1,800 + 1,800 = **23,600**
- **Step 3: Final Billing Amount (Net Total)**
  - Formula: Sub-Total + Roundoff Amount

  - Example: 23,600 + (-0.40) = **23,599.60**
- **Step 4: Calculate Current Balance**
  - Formula: (Opening Balance + Total Bills) - (Total Received + Total TDS)
  - Example: (5,000 + 23,600) - (23,000 + 600) = **5,000**

9. Database Tables Overview

S. No.	Table Name	Description
1	**aggrement_time**	Stores standard predefined rental agreement durations in months (e.g., 1, 11, 12, or 24 months).
2	**agreementfile_tbl**	Used to track and store file names of digital rental agreement documents uploaded for specific tenants.
3	**billing_tbl**	Contains records of all generated rent bills, tracking rent amounts, tax (CGST/SGST), billing periods, and invoice status.
4	**company_tbl**	Stores details of the companies managing the rentals, including contact information, GSTIN, and bank details.
5	**hendler_tbl**	Intended for storing information about "handlers" or agents (currently no data dumped).
6	**master_tbl**	Manages system users and admins, storing encrypted credentials, contact details, and user types.
7	**message_tbl**	A log for tracking messages sent to tenants, including mobile numbers, message content, and timestamps.
8	**owner_tbl**	Stores information about property owners, including PAN/GST numbers and bank accounts for disbursements.
9	**payment_tbl**	Records all rent payments received, tracking receipt numbers, TDS deductions, payment modes, and applicable months.
10	**tenant_tbl**	The central table for tenant personal details, shop numbers, deposits, monthly rent, and agreement dates.

Special Database Objects

In addition to the tables, the script includes the following:

S. No.	Object Type	Name	Description
1	**View**	balance_view_tbl	A virtual table summarizing tenant balances by combining data from tenant, billing, and payment tables.
2	**Stored Procedure**	balance_detail	Calculates the total bill amount and total received amount for a specific tenant.
3	**Stored Procedure**	report_monthPro	Generates detailed monthly reports for a tenant (Rent, GST, and Opening Balances).