WEB

WebStamp CRM - Technical Implementation Guide

This document describes how the WebStamp CRM application is built, including architecture, data model, API routes, authentication and roles, UI structure, background processing, deployment, and troubleshooting. It is based on the current codebase.

1. Architecture Overview

Framework: Laravel 12 (PHP 8.2+)

Architecture: API-first REST + single Blade view for the UI shell

Frontend: Blade template + vanilla JavaScript (Axios) + custom CSS (Tailwind v4 build pipeline)

Auth: Laravel Sanctum token-based auth (Bearer tokens)

RBAC: Role-based access control using role middleware

Storage: Private disk for invoices and branding assets

Queues: Database queue for invoice emails and PDF generation

Scheduler: Daily scheduled command to generate subscription invoices

2. Project Structure (Key Paths)

routes/web.php- Serves the SPA-like Blade app shell

routes/api.php- All JSON API endpoints

app/Http/Controllers- API controllers

app/Models- Eloquent models and relationships

app/Http/Resources- API response shaping

resources/views/app.blade.php- Main UI shell

resources/js/app.js- Frontend logic

resources/css/app.css- UI styles and theme system

database/migrations- Schema definition

app/Jobs+app/Services- Invoice emails and PDF generation

routes/console.php- Scheduler configuration

3. Authentication and Authorization

Authentication

Login:POST /api/auth/loginwithemailandpassword

Returns a Sanctum token:{ token, user }

Frontend stores token inlocalStorageand sendsAuthorization: Bearer

Logout:POST /api/auth/logoutrevokes the current token

Authorization (RBAC)

Middleware aliases defined inbootstrap/app.php:role,permission

Role checks are enforced on API routes viaEnsureRolemiddleware

Roles used:admin,staff,customer

Permissions model exists but is not currently enforced in routes

Role Enforcement Summary

Admin + Staff: full CRUD on customers, jobs, subscriptions, invoices, websites, and revenue stats

Admin only: update brand logo

Customer only: portal endpoints (own jobs, subscriptions, invoices, websites)

4. Database Schema

The schema is defined by migrations and mirrored inwebstamp_crm_schema.sql. Soft deletes are enabled for customers, jobs, subscriptions, invoices, and websites.

Core Tables (High-Level)

users: application users

roles,permissions,role_user,permission_role: RBAC

customers: customer records (optional portal linkage viauser_id)

jobs: one-off work

subscriptions: recurring billing

invoices,invoice_line_items: billing

websites: customer websites and login URLs

files: stored files (PDFs, logos)

brand_settings: logo configuration

user_preferences: theme preference

queue_jobs,job_batches,failed_jobs: background processing

personal_access_tokens: Sanctum tokens

Full DDL (MySQL)

5. API Endpoints

All routes are defined inroutes/api.php. Unless noted, routes requireauth:sanctum.

Auth

POST /api/auth/login(public) - returns token + user

GET /api/auth/me

POST /api/auth/logout

Account (Authenticated)

PUT /api/account/profile- update name/email

PUT /api/account/password- update password

Preferences

GET /api/preferences- get theme

PUT /api/preferences- update theme

Branding

GET /api/brand- current brand settings

GET /api/brand/logo(public) - serve current logo file

POST /api/brand/logo(admin only) - upload logo

Customers (admin, staff)

GET /api/customers- supportssearch,per_page

POST /api/customers

GET /api/customers/{id}

PUT /api/customers/{id}

DELETE /api/customers/{id}

Jobs (admin, staff)

GET /api/jobs- supportscustomer_id,status,per_page

POST /api/jobs

GET /api/jobs/{id}

PUT /api/jobs/{id}

DELETE /api/jobs/{id}

Subscriptions (admin, staff)

GET /api/subscriptions- supportscustomer_id,status,per_page

POST /api/subscriptions

GET /api/subscriptions/{id}

PUT /api/subscriptions/{id}

DELETE /api/subscriptions/{id}

Invoices (admin, staff)

GET /api/invoices- supportscustomer_id,status,per_page

POST /api/invoices- includes line items

GET /api/invoices/{id}

PUT /api/invoices/{id}

DELETE /api/invoices/{id}

POST /api/invoices/{id}/send- generate PDF + send email

GET /api/invoices/{id}/download- download PDF

Websites (admin, staff)

GET /api/websites- supportscustomer_id,per_page

POST /api/websites

GET /api/websites/{id}

PUT /api/websites/{id}

DELETE /api/websites/{id}

Stats (admin, staff)

GET /api/stats/revenue- returns completed jobs total + active subscriptions total

Customer Portal (customer)

GET /api/portal/jobs

GET /api/portal/subscriptions

GET /api/portal/invoices

GET /api/portal/invoices/{id}

GET /api/portal/invoices/{id}/download

GET /api/portal/websites

6. Business Logic Details

Customers

Customers can be linked to a portal user viacustomers.user_id

Customer detail view loads jobs, subscriptions, invoices, and websites

Aggregates (via API): jobs count, subscriptions count, total spent, and monthly recurring revenue

Jobs

Statuses:draft,completed,invoiced

When status becomescompletedorinvoiced, timestamps are set if missing

Subscriptions

Billing frequency currently fixed tomonthly

next_invoice_datedefaults tostart_date

Status:activeorpaused

Invoices

Invoice numbers generated asINV-YYYYMMDD-XXXXXXif not supplied

Line items support manual entries or linking to jobs/subscriptions viabillable_type/billable_id

When a job is linked as a line item, the job is marked asinvoiced

Totals are computed on the server from line items + tax

Revenue Calculation

7. Background Jobs and Scheduling

Job:SendInvoiceEmail(queued)

PDF Generation:InvoicePdfServiceuses DomPDF

Recurring Invoices:invoices:generate-recurring

Schedule: daily at 02:00, configured inroutes/console.php

Auto-send recurring invoices controlled byINVOICES_AUTO_SEND_RECURRING=true

8. Storage and Files

Private disk:storage/app/private

Invoices and logos are stored as private files and served via controller

Brand logo is served via/api/brand/logo(public)

9. UI / UX Implementation

App Shell

Single Blade view with multiple sections (data-view)

Navigation updates active view and triggers data loads via JS

Logo area links to Admin page

Theme System

CSS variables define theme colors and accents

Light and Dark themes toggle viadata-themeon

User preference stored inuser_preferencesand localStorage

Responsive Behavior

Mobile menu at <= 964px with full-screen overlay

Customer cards hide columns at <= 725px

Logout button moves into menu at <= 450px

Formatting

Currency formatted as GBP (en-GB)

Dates formatted for display (short format)

10. Deployment

Environment Setup

Configure.env: DB_*, APP_URL, MAIL_*, QUEUE_CONNECTION=database

SetINVOICES_AUTO_SEND_RECURRINGif desired

Build and Run

Queues

Scheduler (Production Cron)

11. Troubleshooting

Revenue Box Shows "--"

Check network call to/api/stats/revenue

Ensure auth token is set and API route is accessible

Clear config/route cache after deploy

Logo Not Showing

Ensure a logo file exists inbrand_settings.logo_file_id

Confirm/api/brand/logoreturns 200

Check storage permissions forstorage/app/private

No Emails / PDFs Sent

Queue worker not running

Mail configuration missing in.env

Assets Not Updating

Rebuildnpm run build

Deploypublic/buildandpublic/manifest.json

12. Security Notes

Passwords are hashed via Laravel

Portal data is restricted via role middleware and customer linkage

Files are stored on the private disk and served via controller responses

13. Future Improvements (Optional)

Monthly revenue with date filtering and invoice status filtering

Permissions-based authorization enforcement

Granular audit logs for billing changes

Multi-currency support

Public customer portal subdomain

Document version:2026-02-06