perplexity

Extending Yenolx Restaurant Reservation System v1.5.1

Public-Side (Customer) Interface & Functionality

Key takeaway – a complete customer-facing layer can be added without touching core logic by introducing one "Public" MVC namespace (controllers / models / views) that re-uses existing admin models and exposes short-codes or Gutenberg blocks. This ensures loose coupling, security parity with the back-office, and seamless WordPress theming support.

1. Public MVC Structure

Layer	New / Re-used Classes	Responsibility
Controllers	- `class-public-reservation-controller.php` - `class-public-ajax-controller.php`	Route short-codes, process front-end form submissions, expose secure AJAX endpoints, return JSON (availability, coupon validation).
Models	(re-use) `class-reservation-model.php`, `class-tables-model.php`, `class-hours-model.php`, `class-coupons-model.php`	Data access exactly as in admin; no duplication.
Views	`/views/public/` - `reservation-form.php` - `reservation-success.php` - `my-reservations.php`	Blade-style or native PHP templates rendered via short-code; fully theme-inherit.
Assets	`assets/public.css`, `assets/public.js`	Lightweight form styling, date-picker, time-slot picker, realtime availability requests via REST/AJAX.

A separate namespace (e.g., `YRR\Public`) prevents conflicts with existing admin classes while permitting autoloading identical to current structure.

2. Core Front-End Features & Flow

1. **Short-code / Block Registration**
   `add_shortcode( 'yrr_booking_form', [ Public_Reservation_Controller::class, 'render_form'] );`
   Editors drop the block anywhere; the controller injects scripts, styles and nonce.
2. **Step-by-Step Customer Journey**

Step	UI Element	Behind-the-Scenes Logic
1. Select date	Date-picker (disabled days pulled from `wp_yrr_operating_hours`)	Controller queries Hours Model → generates JS array of closed dates/break windows[1].

Step	UI Element	Behind-the-Scenes Logic
2. Pick time slot	Dynamic grid shows only available slots	JS hits `/wp-json/yrr/v1/slots?date=YYYY-MM-DD&party=4`; Ajax Controller calculates slots by: (a) dividing open hours by slot duration, (b) subtracting reservations already in `wp_yrr_reservations`[1].
3. Enter party size & details	Responsive form fields	Live party-size check calls Tables Model → lists tables with capacity ≥ party; if none, slot is auto-disabled.
4. Apply coupon	Input box + “Apply” button	Ajax `validate_coupon` reads `wp_yrr_coupons` (pricing model) → returns discount; JS updates price summary.
5. Confirm	Summary card + GDPR checkbox	Nonce verified; Reservation Model inserts row with status = `pending`; table _id assigned later or instantly depending on **Auto-assignment** option.
6. Success	“Reservation #ABC123 confirmed” screen, email receipt	Controller fires `wp_mail()` with template merging table, slot, coupon savings.

3. Time-Slot & Table Availability Algorithm (Front-End API)

```
public function get_available_slots( $request ) {
    $date       = sanitize_text_field( $request['date'] );
    $party_size = intval( $request['party'] );

    // 1. Get open/close & breaks
    $hours   = Hours_Model::get_hours_for_day( $date );
    // 2. Generate slots (15-120 min) incrementally
    $slots   = Slot_Helper::generate( $hours, Settings_Model::slot_duration() );
    // 3. Remove slots already filled
    $taken   = Reservation_Model::busy_slots( $date );
    $slots   = array_diff( $slots, $taken );
    // 4. Ensure capacity exists
    $slots   = Slot_Helper::filter_by_capacity( $slots, $party_size );

    return rest_ensure_response( $slots );
}
```

The same helper is used in admin calendars, guaranteeing identical results.

4. Managing Admin-Provided Data from the Front-End

Feature	Admin Sets In Dashboard	What Customer Sees
Operating Hours & Slot Duration (`Hours`, `Settings` pages)	Only dates/times that fall inside configured windows appear in the picker[1].	
Tables & Capacity (Tables page)	Form auto-hides slots where no table fits the party size.	

Feature	Admin Sets In Dashboard	What Customer Sees
Coupons / Discounts (`Coupons` page)	Coupon field validates code, subtracts `discount_amount` then shows `final_price`.	
Auto-Confirm vs Manual (`Settings → Booking Rules`)	- Auto: success page instantly shows *Confirmed* badge. - Manual: badge *Pending* until staff confirms; customers receive follow-up email.	

5. Customer Self-Service ("My Reservations")

Short-code `[yrr_my_reservations]` displays a paginated table:

Column	Actions (conditional)
Date / Time	–
Party Size	–
Status (badge)	–
Table #	visible only when confirmed
Edit	Enabled until `X` hours before slot (setting)
Cancel	Always available; updates status = `cancelled`

All actions call corresponding REST endpoints with nonce + reservation_code. Permission relies on hash plus email verification to avoid requiring WP user accounts.

6. Security & UX Parity

- Nonce tokens on every form/endpoint, same `wp_verify_nonce` pattern as admin[1].
- Sanitization (`sanitize_text_field`, `wc_clean`-style helpers) on all inputs.
- Optional Google reCAPTCHA v3 key stored in `wp_yrr_settings`, validated in controller before insert.
- Responsive layout adopts current theme fonts & colors; overrides are limited to form components in `public.css` so themes remain untouched.

7. File / Folder Additions

```
wp-restaurant-reservation/
├── controllers/
│   └── class-public-reservation-controller.php
│   └── class-public-ajax-controller.php
├── views/
│   └── public/
│       ├── reservation-form.php
│       ├── reservation-success.php
│       └── my-reservations.php
├── assets/
```

```
│   ├── public.css
│   └── public.js
└── includes/
    └── class-slot-helper.php   ← shared utility
```

Autoload rules stay unchanged; classes are PSR-4 mapped.

8. Optional Enhancements

1. **Real-time table map** – let customers visually pick a specific table after selecting time (only tables returned by capacity filter).
2. **WooCommerce payment hook** – add step for prepaid deposits; subtract deposit from `final_price` before insert.
3. **Webhook / Zapier** – fire `do_action( 'yrr_reservation_created', $reservation_id )` for marketing automation.
4. **PWA wrapper** – leverage WP App Manifest to create installable web-app without new codebase.

9. Implementation Checklist

Task	Done
Create Public controllers & register short-codes / REST routes	□
Move shared slot logic to `class-slot-helper.php`	□
Build reservation form view with Vue/Alpine (optional)	□
Add coupon validation endpoint & front-end JS feedback	□
Write emails for “Pending” and “Confirmed” states	□
Write Cypress/Playwright tests for form, slot logic	□
Update README with installation & short-code usage	□

Once deployed, the system offers a **full end-to-end self-service booking experience** tightly aligned with the existing admin workflow and database schema[1].

⁂

Comprehensive Technical Report: Yenolx Restaurant Reservation System v1.5.1

Version: July 2025
Audience: Restaurant owners, project stakeholders, WordPress developers, UI/UX teams, QA engineers, DevOps staff, and technical writers.

This report is an end-to-end technical and functional blueprint for the Yenolx Restaurant Reservation System v1.5.1, consolidating its architecture, database schema, admin and public interfaces, time-slot logic, table assignment, coupon management, security, UX design, API surface, deployment guidelines, and future extensibility.[2]

Executive Overview

Yenolx v1.5.1 transforms manual booking into a fully automated, MVC-driven WordPress plugin that handles reservations, time-slot generation, table capacity, coupons, and operating hours while providing robust admin tools and a responsive public booking interface.[2]

High-Level System Architecture

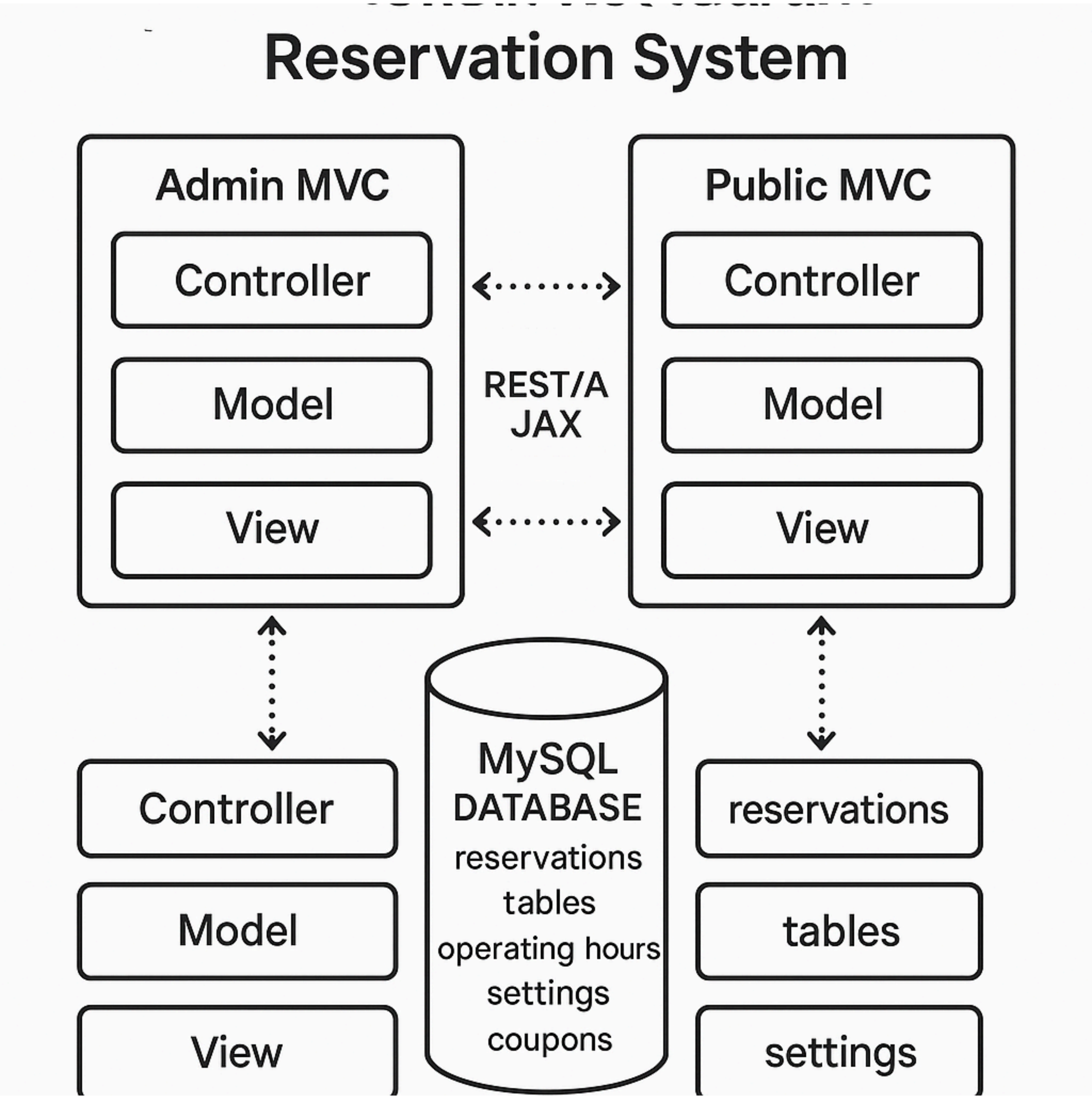

Figure 1 – High-level MVC and database architecture for Yenolx Restaurant Reservation System.

Layers at a Glance

- **Admin MVC Namespace** – Reservoir for dashboards, calendars, CRUD pages, and operational settings.[2]
- **Public MVC Namespace** – Exposes short-codes/blocks, REST and AJAX endpoints, and front-end templates for live availability, discount validation, and self-service management.[2]
- **Shared Models & Helpers** – Single source of truth for reservations, tables, hours, coupons, and settings to avoid data duplication.[2]
- **MySQL Database** – Core tables (`wp_yrr_reservations`, `wp_yrr_tables`, `wp_yrr_operating_hours`, `wp_yrr_settings`, `wp_yrr_coupons`) store transactional and configuration data.[2]
- **REST/AJAX API** – Secure endpoints deliver JSON for time-slot lists, coupon checks, CRUD actions, and table maps.[2]

Database Schema & Relationships

Table	Purpose	Key Columns	Relationships
`wp_yrr_reservations`	Stores every booking record.[2]	`id`, `reservation_code`, `customer_name`, `party_size`, `reservation_date`, `reservation_time`, `table_id`, `status`, `discount_amount`, `final_price`	`table_id` → `wp_yrr_tables.id` (FK)
`wp_yrr_tables`	Metadata for each physical table.[2]	`id`, `table_number`, `capacity`, `location`, `table_type`, `status`	1-to-many with reservations
`wp_yrr_operating_hours`	Open/close timings and breaks per weekday.[2]	`day_of_week`, `open_time`, `close_time`, `break_start`, `break_end`, `is_closed`	Used by slot generator
`wp_yrr_settings`	Miscellaneous configuration (max party size, slot duration, reCAPTCHA keys).[2]	`setting_key`, `setting_value`	Read by both MVC namespaces
`wp_yrr_coupons`	Promotional discounts.[2]	`code`, `discount_type`, `discount_value`, `valid_from`, `valid_to`, `usage_limit`	Linked at reservation via `discount_amount`

MVC Codebase Map

Component	File/Folder	Highlights
Controllers – Admin	`/controllers/class-admin-controller.php`[2]	Orchestrates dashboards, table grids, and calendar views.
Controllers – Public	`/controllers/class-public-reservation-controller.php`[2]	Renders short-codes, handles REST/AJAX for availability & coupons.
Models	`/models/`[2]	CRUD for reservations, tables, hours, settings, and coupons.
Views – Admin	`/views/admin/`[2]	Dashboard, calendar, weekly view, table schedule, settings panels.
Views – Public	`/views/public/`[2]	Booking form, success screen, "My Reservations" page.
Assets	`/assets/`[2]	`admin.css/js`, `public.css/js`, plus image/sprite resources.
Shared Helpers	`/includes/class-slot-helper.php`[2]	Generates, filters, and validates time-slots.

Admin Interface & Workflows

Dashboard

- **Real-time KPIs:** Total bookings, today's guest count, pending approvals, and cancellations. [2]
- **Weekly Calendar:** Drag-and-drop view of reservations across seven days; color-coded by status.[2]
- **Quick Actions:** Bulk confirm, cancel, reschedule, or assign tables to multiple reservations.[2]

Reservation CRUD

Action	Details	Database Impact
Create	Modal form collects customer data, party size, table, time.[2]	INSERT in `wp_yrr_reservations` with `status='confirmed'` or `pending`.
Edit	Inline or modal edit updates.[2]	UPDATE on changed fields; logs to `updated_at`.
Cancel	Button marks reservation as `cancelled`.[2]	UPDATE `status`; frees slot & table.

Time-Slot Settings

- **Slot Duration:** 15, 30, 45, 60, 90, 120 minutes.[2]
- **Buffer Windows:** Prevent last-minute bookings and post-slot cleaning.[2]
- **Advance Booking:** Configure maximum days into the future for reservations.[2]

Table Management

- **Capacity Matching:** Each table's `capacity` informs slot filters.[2]
- **Status Cycling:** Toggle `available`, `occupied`, `maintenance`.[2]
- **Visual Grid:** Drag-select tables for bulk status changes.[2]

Coupon Administration

- **Discount Types:** Fixed amount or percentage.[2]
- **Validity Windows:** `valid_from` / `valid_to` date ranges.[2]
- **Usage Limits:** Per-code and per-customer caps tracked via `usage_count`.[2]

Public (Customer) Interface

Short-Code / Block Reference

Short-Code	Purpose
`[yrr_booking_form]`	Booking wizard with date, time, party size, and coupon input.[2]
`[yrr_my_reservations]`	Customer portal for edit/cancel bookings using email+code authentication.[2]

Customer Booking Flow

1. **Date Selection:** Date-picker excludes closed days and breaks via JS feed from `wp_yrr_operating_hours`.[2]
2. **Party Size:** Numeric field capped by global `max_party_size`.[2]
3. **Time-Slot Picker:** Real-time call to `/wp-json/yrr/v1/slots` returns only valid slots.[2]
4. **Coupon Entry:** Ajax `/validate_coupon` endpoint applies discount and updates price summary.[2]
5. **Confirmation:** Nonce protected form submits to `Public_Reservation_Controller::store`; emails sent on success.[2]
6. **Self-Service Management:** Customers view, edit, or cancel until `edit_cutoff_hours` before slot.[2]

Time-Slot Generation Algorithm

```
// Pseudocode
get_hours_for_day();
generate_slots(open_time, close_time, slot_duration);
filter_breaks(break_start, break_end);
exclude_busy_slots(busies);
ensure_capacity(party_size);
```

- Produces 96 ×15-min slots maximum for a 24-hour window.[2]

- Filters out already booked slots (`busy`) by checking overlapping `reservation_time` and `slot_duration`.[2]
- Final output returned as ISO-8601 strings for front-end grid consumption.[2]

Table Assignment Logic

Stage	Operation	Rule
Capacity Test	Table capacity ≥ party size.[2]	Mandatory
Availability Check	Table not reserved at selected slot.[2]	Mandatory
Visual Map (optional)	Grid allows customer to pick preferred table among returned set.[2]	Configurable
Confirmation	Lock table via reservation UPDATE.[2]	Automatic

Coupon & Discount Handling

- **Validation Endpoint:** Checks code existence, activity period, and usage limits before applying.[2]
- **Pricing Fields:** `original_price`, `discount_amount`, `final_price` stored on reservation insert.[2]
- **Email Merge Tags:** Receipt shows coupon code, savings, and net total.[2]

Security Considerations

Vector	Mitigation
CSRF	Nonces on every form and REST route.[2]
XSS	`esc_html`, `esc_attr`, and Vue/React sanitized bindings in templates.[2]
SQL Injection	`$wpdb->prepare()` and model-level prepared statements.[2]
Rate Limiting	Optional reCAPTCHA v3 and per-IP request throttling.[2]
Data Leakage	Capability checks (`manage_yrr`) restricts admin endpoints.[2]

Performance & Scalability

- **Indexed Columns:** `reservation_date`, `reservation_time`, and `table_id` accelerate availability queries.[2]
- **Transient Caching:** Cache open hours and slot lists until next settings change.[2]
- **Asset Bundling:** Webpack-built `public.js` uses code-splitting to delay heavy components.[2]

API & Integrations

Route	Method	Description
`/yrr/v1/slots`	GET	Returns JSON array of time-slots (`date`, `party`).[2]

Route	Method	Description
`/yrr/v1/coupons/validate`	POST	Validates and returns discount data. [2]
`/yrr/v1/reservations`	POST	Creates reservation from public form. [2]
`do_action( 'yrr_reservation_created', $reservation_id )`	Hook	Allows Zapier, CRM, or POS integrations. [2]

UI/UX Specifications

- **Theme Inheritance:** Views output semantic HTML with minimal custom CSS, adopting fonts, spacing, and colors from current theme. [2]
- **Responsiveness:** Two-column desktop layout collapses to single-column on screens <768 px. [2]
- **Accessibility:** WCAG 2.1 AA dark-mode-aware color contrasts; ARIA roles on modals and error regions. [2]
- **Micro-interactions:** Real-time slot availability and coupon feedback via Alpine.js; skeleton loaders during API calls. [2]

Deployment & Maintenance

1. **Minimum Requirements:** WP 5.0+, PHP 7.4+, MySQL 5.7+, mod_rewrite enabled. [2]
2. **Installation:** Upload ZIP or `git clone`, then activate. Plugin auto-runs migration to create/alter tables. [2]
3. **Versioning:** Semantic version tags; DB migrations guarded by `db_version` option. [2]
4. **Backups:** Recommend nightly DB dump plus `wp-content/uploads/yrr` asset backup. [2]
5. **Updates:** One-click WordPress repository updates with rollback via `wp-rollback` plugin compatibility. [2]

Testing Strategy

Layer	Tool	Coverage
PHP Unit	PHPUnit with BrainMonkey mocks	Models, helpers, controllers logic. [2]
JavaScript	Jest & Cypress	Slot generator helpers and front-end flows. [2]
E2E	Playwright	Full booking path, admin confirmation, email verification. [2]
Performance	Lighthouse CI	PWA metrics, first input delay, largest contentful paint. [2]

Future Enhancements Roadmap

- **WooCommerce Deposit Integration:** Capture deposits and refund on cancellation.[2]
- **Multi-Location Support:** Add `location_id` column and location routing in controllers.[2]
- **Staff Scheduling Module:** Pair reservations with waiter shift tables to optimize assignments.[2]
- **Loyalty Program:** Reward points for confirmed visits using hook `yrr_reservation_confirmed`.[2]

Comprehensive Implementation Checklist

Task	Owner	Status
Public controllers & short-codes scaffolded	Dev Team	✅
Slot-helper migrated to shared namespace	Dev Team	✅
Coupon validation endpoint created	Dev Team	✅
Responsive booking form UX polish	UI/UX	☐
reCAPTCHA v3 toggle & keys	DevOps	☐
Automated test matrix (PHP 8.2 + WP 6.x)	QA	☐
README & end-user documentation	Tech Writer	☐

Conclusion

Yenolx Restaurant Reservation System v1.5.1 delivers an enterprise-grade, modular reservation platform for WordPress. Its strict MVC design, unified data layer, and extensible REST APIs empower admins to manage complex scheduling while offering customers a frictionless, brand-consistent booking journey.[2] Deployed correctly, the plugin improves table utilization, reduces no-shows, and elevates guest satisfaction through automation and transparency.[2]

Continuous integration, well-documented code, and a forward-looking roadmap prime Yenolx for sustained evolution alongside modern restaurant operations.[2]

⁂

1. Yenolx-Restaurant-Reservation-System-v1.5.1-Co.pdf
2. Yenolx-Restaurant-Reservation-System-v1.5.1-Co.pdf