

NovaRax Multi-Tenant SaaS Platform

Complete Technical Documentation & Implementation Guide

Version: 1.0

Date: October 31, 2025

Project Duration: 4 Months

Prepared For: NovaRax Development Team & Board of Directors

Table of Contents

1. [Executive Summary](#)
 2. [Project Overview](#)
 3. [System Architecture](#)
 4. [Technical Infrastructure](#)
 5. [Database Architecture](#)
 6. [Security & Authentication](#)
 7. [Feature Specifications](#)
 8. [Development Phases](#)
 9. [Team Structure & Responsibilities](#)
 10. [Technology Stack](#)
 11. [Risk Management](#)
 12. [Success Metrics](#)
 13. [Future Roadmap](#)
-

1. Executive Summary

1.1 Project Vision

NovaRax is building a modern, scalable multi-tenant SaaS platform powered by WordPress, offering modular business applications (CRM, Money Manager, Loan Management, AI Assistant) through a subscription-based model.

1.2 Business Model

- **Subscription-based Revenue:** Monthly/annual recurring revenue per module
- **Multi-tenant Architecture:** One platform serving multiple isolated client instances
- **Modular System:** Clients purchase only the modules they need
- **White-label Solution:** Fully branded experience with no WordPress visibility

1.3 Key Differentiators

- ✔ **Database Isolation:** Complete data separation per tenant for security and compliance
- ✔ **Seamless SSO:** Single sign-on across admin and tenant dashboards
- ✔ **Dynamic Provisioning:** Automated subdomain and database creation
- ✔ **Flexible Subscriptions:** Pay-as-you-go module activation
- ✔ **Enterprise-Grade:** Built for scalability and professional use

1.4 Target Market

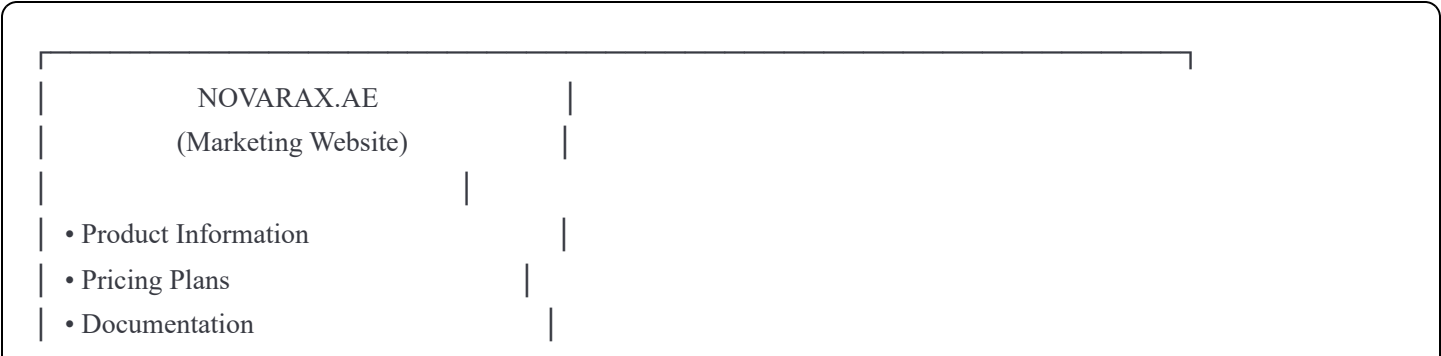
- **Primary:** Small to medium businesses (10-100 employees)
- **Secondary:** Startups and entrepreneurs
- **Tertiary:** Enterprise clients requiring isolated instances

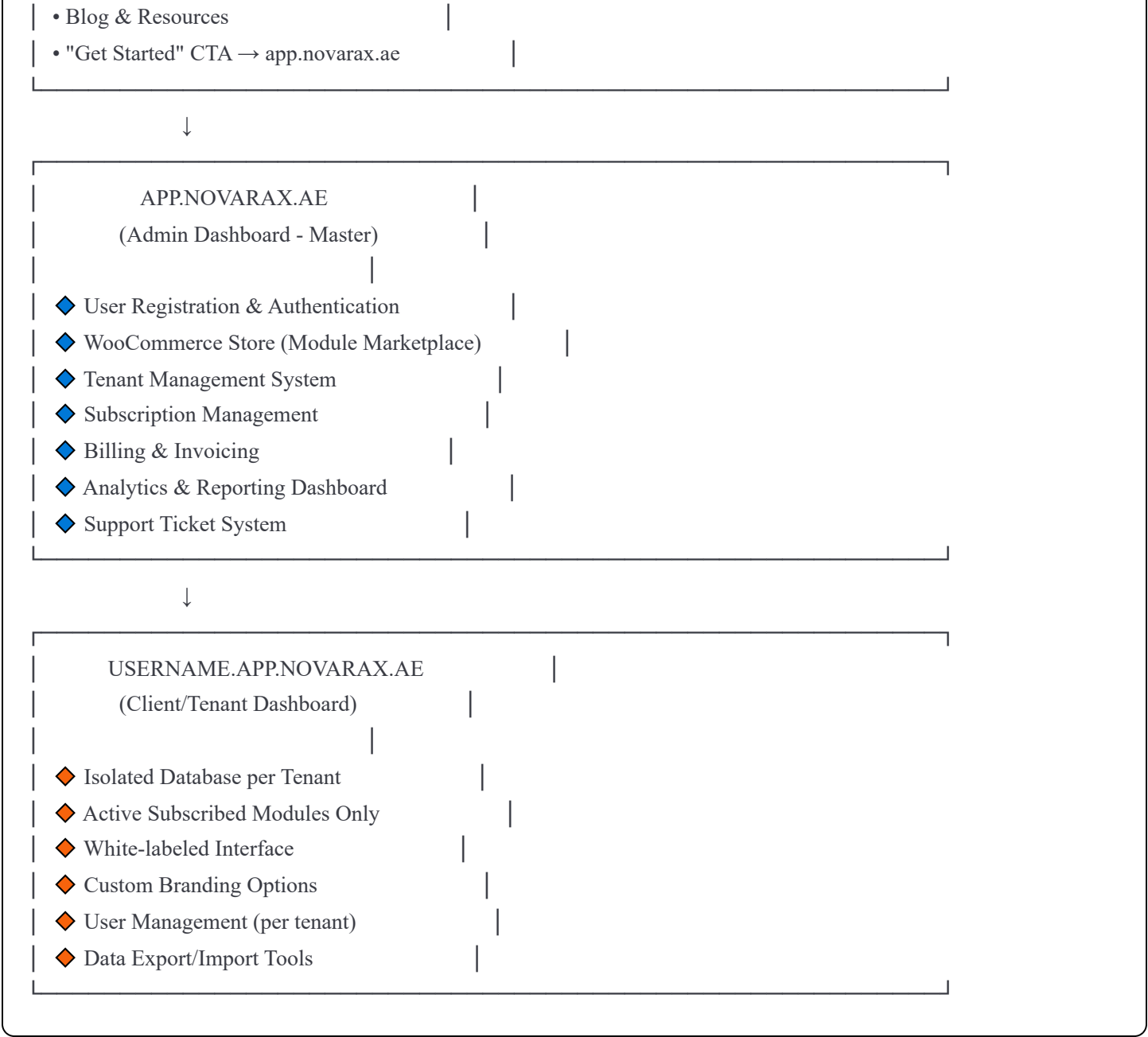
1.5 Success Criteria

- Launch with 4 core modules within 4 months
- Support 100+ concurrent tenants
- 99.9% uptime SLA
- <3 second page load times
- Zero data breach incidents

2. Project Overview

2.1 System Components





2.2 User Journey Flow

Phase 1: Discovery & Registration

- 1. User visits novarax.ae (marketing site)
- 2. Explores available modules and pricing
- 3. Clicks "Get Started" button
- 4. Redirected to app.novarax.ae/register

Phase 2: Account Creation

- 5. Fills registration form:
 - Full Name

- Company Name
- Email Address
- Phone Number
- Desired Username (subdomain)
- Password
- Billing Address

6. System validates username availability
7. Creates tenant record in master database
8. Creates WordPress user with "Tenant" role
9. Sends welcome email with credentials

Phase 3: Module Selection

10. Auto-login to `app.novarax.ae/apps`
11. Browse module marketplace
12. Add desired modules to cart
13. Proceed to checkout
14. Complete payment via WooCommerce

Phase 4: Provisioning

15. System automatically:
 - Creates subdomain: `username.app.novarax.ae`
 - Provisions isolated database
 - Deploys WordPress codebase
 - Activates purchased modules
 - Configures white-label branding
16. Sends onboarding email with dashboard link

Phase 5: Usage

17. User accesses `username.app.novarax.ae`
18. Uses subscribed modules

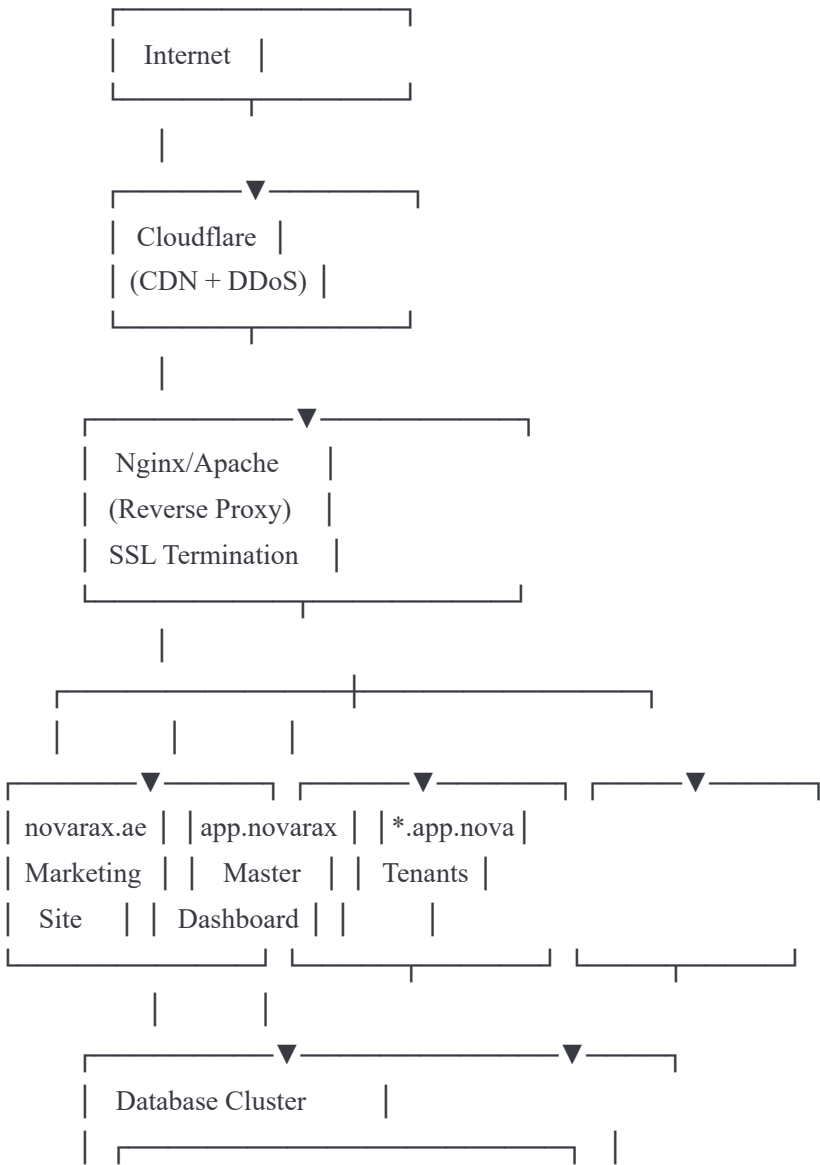
- 19. Can return to app.novarax.ae/apps to add more modules
- 20. Single authentication works across both domains

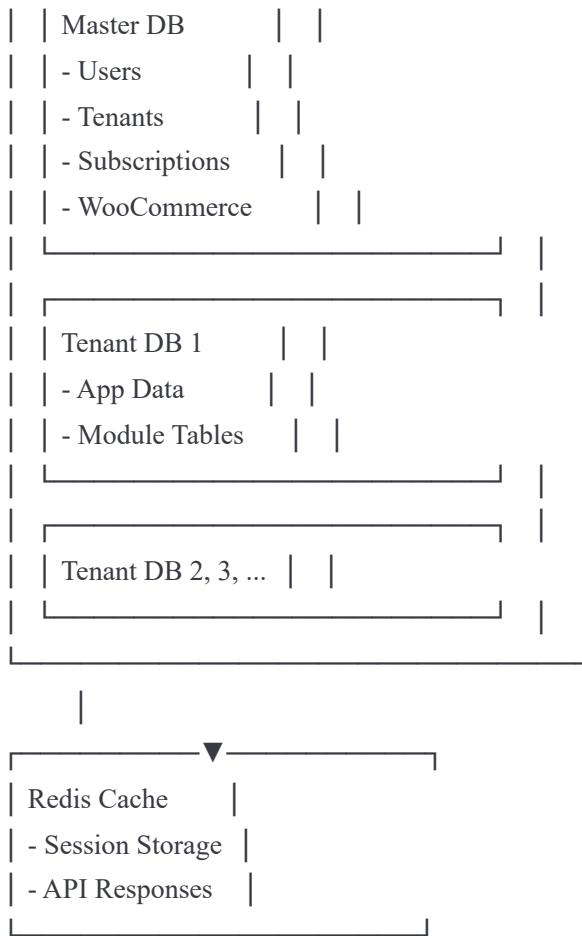
Phase 6: Ongoing Management

- 21. Subscription auto-renews monthly/annually
- 22. Modules deactivate if subscription lapses
- 23. User can upgrade/downgrade plans
- 24. Admin monitors usage and provides support

3. System Architecture

3.1 High-Level Architecture Diagram





3.2 WordPress Installation Structure

```

/var/www/
|
├── novarax.ae/           # Marketing Website
|   ├── wp-admin/
|   ├── wp-content/
|   └── wp-config.php
|
├── app.novarax.ae/       # Master/Admin Dashboard
|   ├── wp-admin/
|   ├── wp-content/
|   │   ├── mu-plugins/   # Must-Use Plugins
|   │   │   ├── novarax-tenant-manager/
|   │   │   ├── novarax-sso/
|   │   │   ├── novarax-api/
|   │   │   └── novarax-license-server/
|   │   └── plugins/
|   │       ├── woocommerce/
|   │       ├── woocommerce-subscriptions/
|   │       └── white-label-cms/
  
```

```

| | └─ themes/
| |   └─ novarax-admin-theme/
| └─ wp-config.php
|
└─ tenant-dashboard/      # Shared Tenant Codebase
    └─ wp-admin/
    └─ wp-content/
        └─ mu-plugins/
            └─ novarax-tenant-bootstrap/
            └─ novarax-license-client/
            └─ novarax-sso-client/
        └─ plugins/      # Modular Apps
            └─ novarax-money-manager/
            └─ novarax-crm/
            └─ novarax-ai-assistant/
            └─ novarax-loan-management/
        └─ themes/
            └─ novarax-tenant-theme/
    └─ wp-config.php      # Dynamic config

```

3.3 Component Responsibilities

Marketing Website (novarax.ae)

- Showcase products and features
- SEO-optimized content
- Lead capture forms
- Redirect to app.novarax.ae for registration

Master Dashboard (app.novarax.ae)

- User authentication hub
- WooCommerce storefront
- Tenant CRUD operations
- Subscription lifecycle management
- Global analytics and reporting
- Support ticket system
- Billing and invoicing

Tenant Dashboard (username.app.novarax.ae)

- Application workspace
 - Module interfaces
 - Data management
 - User settings
 - Usage analytics
 - Tenant-specific customization
-

4. Technical Infrastructure

4.1 Server Requirements

Minimum Specifications (Launch)

- **Server Type:** VPS/Cloud Instance (IONOS VPS)
- **RAM:** 16 GB (as planned)
- **CPU:** 4 vCPU cores
- **Storage:** 200 GB SSD
- **Bandwidth:** Unmetered or 10TB+
- **OS:** Ubuntu 22.04 LTS or CentOS Stream 9

Recommended Specifications (Production)

- **RAM:** 32 GB
- **CPU:** 8 vCPU cores
- **Storage:** 500 GB SSD (with auto-scaling)
- **Backup Storage:** 1 TB separate volume

Scaling Plan (Year 1)

Tenant Count	RAM	CPU	Storage	Estimated Cost
0-50	16GB	4	200GB	\$80-120/mo
51-150	32GB	8	500GB	\$150-250/mo
151-500	64GB	16	1TB	\$300-500/mo
500+	Load Balancer + Multiple Servers	\$800+/mo		

4.2 DNS Configuration

Wildcard DNS Setup

Type	Name	Value	TTL
A	novarax.ae	123.456.789.100	3600
A	app.novarax.ae	123.456.789.100	3600
A	*.app.novarax.ae	123.456.789.100	3600
CNAME	www	novarax.ae	3600

Purpose:

- *.app.novarax.ae allows any subdomain (tenant1, tenant2, etc.) to automatically resolve to your server
- No need to manually create DNS records for each tenant

4.3 SSL/TLS Configuration

Wildcard SSL Certificate (IONOS Add-on)

- **Certificate Type:** Wildcard SSL
- **Coverage:** *.app.novarax.ae + app.novarax.ae + novarax.ae
- **Provider:** IONOS or Let's Encrypt
- **Renewal:** Automatic every 90 days
- **Protocol:** TLS 1.3

Certificate Installation Locations

- Nginx/Apache SSL configuration
- All subdomains automatically covered
- Force HTTPS redirection on all domains

4.4 Web Server Configuration

Nginx Configuration (Recommended)

Master Virtual Host:

```
server {  
    listen 80;  
    server_name app.novarax.ae;  
    return 301 https://$server_name$request_uri;  
}  
  
server {  
    listen 443 ssl http2;  
    server_name app.novarax.ae;  
  
    root /var/www/app.novarax.ae;  
    index index.php;  
  
    ssl_certificate /path/to/wildcard.crt;  
    ssl_certificate_key /path/to/wildcard.key;  
  
    # WordPress permalinks  
    location / {  
        try_files $uri $uri/ /index.php?$args;  
    }  
  
    location ~ \.php$ {  
        fastcgi_pass unix:/var/run/php/php8.2-fpm.sock;  
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;  
        include fastcgi_params;  
    }  
}
```

Wildcard Tenant Virtual Host:

```

server {
    listen 443 ssl http2;
    server_name ~^(?<subdomain>.+)\.app\.novarax\.ae$;

    root /var/www/tenant-dashboard;
    index index.php;

    ssl_certificate /path/to/wildcard.crt;
    ssl_certificate_key /path/to/wildcard.key;

    # Pass subdomain to WordPress
    location / {
        try_files $uri $uri/ /index.php?$args;
    }

    location ~ \.php$ {
        fastcgi_pass unix:/var/run/php/php8.2-fpm.sock;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        fastcgi_param HTTP_HOST $host;
        include fastcgi_params;
    }
}

```

4.5 PHP Configuration

Required PHP Settings

```

ini

memory_limit = 512M
upload_max_filesize = 256M
post_max_size = 256M
max_execution_time = 300
max_input_vars = 5000
max_input_time = 300

```

Required PHP Extensions

- php-fpm
- php-mysql
- php-curl

- php-gd
- php-mbstring
- php-xml
- php-zip
- php-imagick
- php-redis (for caching)

4.6 Database Server Configuration

MySQL 8.0 Settings

```
ini

[mysqld]
max_connections = 500
innodb_buffer_pool_size = 8G (50% of RAM)
innodb_log_file_size = 512M
query_cache_size = 0
query_cache_type = 0
max_allowed_packet = 256M
```

Database Naming Convention

- Master: `novarax_master`
- Tenants: `novarax_tenant_username`

4.7 Redis Cache Configuration

Purpose

- Session storage (shared between domains)
- API response caching
- Database query caching
- Object caching for WordPress

Configuration

maxmemory 2gb
maxmemory-policy allkeys-lru
save ""
appendonly yes

4.8 Backup Strategy

Automated Backup Schedule

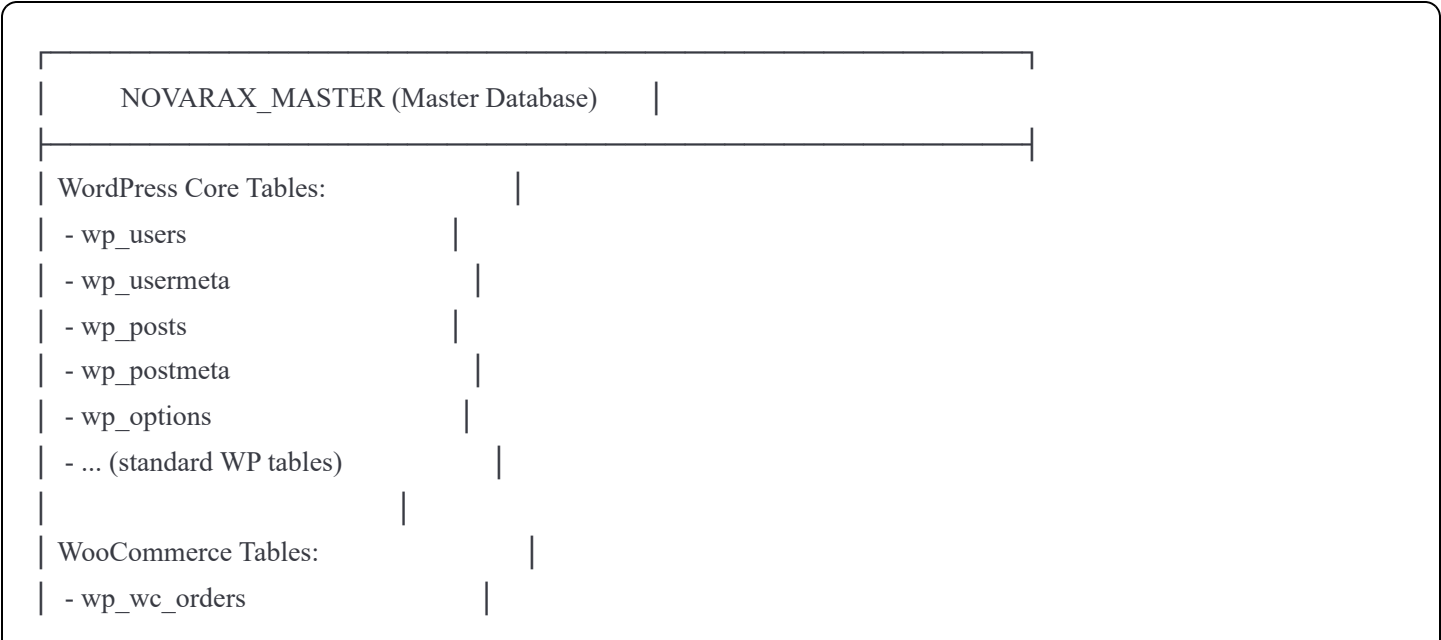
Backup Type	Frequency	Retention	Storage Location
Database Full	Daily 2 AM	30 days	Separate volume
Database Incremental	Every 6 hours	7 days	Separate volume
File System	Weekly	4 weeks	Cloud storage
Configuration	On change	90 days	Version control

Disaster Recovery Plan

- **RTO (Recovery Time Objective):** 4 hours
- **RPO (Recovery Point Objective):** 1 hour
- **Backup Testing:** Monthly restoration drills
- **Off-site Storage:** AWS S3 or Backblaze B2

5. Database Architecture

5.1 Database Structure Overview



- wp_wc_order_items	
- wp_wc_subscriptions	
- wp_wc_products	
- ... (WooCommerce tables)	
Custom NovaRax Tables:	
- wp_novarax_tenants	
- wp_novarax_tenant_subscriptions	
- wp_novarax_modules	
- wp_novarax_tenant_modules	
- wp_novarax_api_keys	
- wp_novarax_audit_logs	
- wp_novarax_usage_stats	

NOVARAX_TENANT_USERNAME (Per-Tenant DBs)	
--	--

WordPress Core Tables:	
- wp_users (tenant users only)	
- wp_usermeta	
- wp_posts	
- wp_postmeta	
- wp_options	
- ... (standard WP tables)	
Module-Specific Tables:	
Money Manager:	
- wp_mm_transactions	
- wp_mm_accounts	
- wp_mm_categories	
CRM:	
- wp_crm_contacts	
- wp_crm_companies	
- wp_crm_deals	
- wp_crm_activities	
Loan Management:	
- wp_loan_applications	
- wp_loan_payments	
- wp_loan_schedules	
AI Assistant:	

- wp_ai_conversations	
- wp_ai_prompts	
- wp_ai_usage_logs	

5.2 Custom Table Schemas

wp_novarax_tenants

sql

```
CREATE TABLE wp_novarax_tenants (  
  id BIGINT(20) UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
  user_id BIGINT(20) UNSIGNED NOT NULL,  
  tenant_username VARCHAR(100) NOT NULL UNIQUE,  
  account_name VARCHAR(255) NOT NULL,  
  company_name VARCHAR(255),  
  subdomain VARCHAR(100) NOT NULL UNIQUE,  
  database_name VARCHAR(100) NOT NULL,  
  status ENUM('active', 'suspended', 'cancelled', 'pending') DEFAULT 'pending',  
  storage_used BIGINT(20) DEFAULT 0,  
  storage_limit BIGINT(20) DEFAULT 5368709120, -- 5GB in bytes  
  user_limit INT DEFAULT 10,  
  phone_number VARCHAR(50),  
  address TEXT,  
  billing_email VARCHAR(255),  
  created_at DATETIME DEFAULT CURRENT_TIMESTAMP,  
  last_login DATETIME,  
  metadata LONGTEXT, -- JSON for flexible data  
  
  INDEX idx_user_id (user_id),  
  INDEX idx_subdomain (subdomain),  
  INDEX idx_status (status),  
  FOREIGN KEY (user_id) REFERENCES wp_users(ID) ON DELETE CASCADE  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

wp_novarax_modules

sql

```
CREATE TABLE wp_novarax_modules (  
  id BIGINT(20) UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
  module_name VARCHAR(255) NOT NULL,  
  module_slug VARCHAR(100) NOT NULL UNIQUE,  
  plugin_path VARCHAR(255) NOT NULL, -- e.g., 'novarax-crm/novarax-crm.php'  
  product_id BIGINT(20), -- WooCommerce product ID  
  description TEXT,  
  version VARCHAR(50),  
  requires_php VARCHAR(10),  
  requires_modules TEXT, -- JSON array of dependent modules  
  icon_url VARCHAR(500),  
  status ENUM('active', 'inactive', 'deprecated') DEFAULT 'active',  
  created_at DATETIME DEFAULT CURRENT_TIMESTAMP,  
  updated_at DATETIME ON UPDATE CURRENT_TIMESTAMP,  
  
  INDEX idx_slug (module_slug),  
  INDEX idx_product (product_id)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

wp_novarax_tenant_modules

sql


```

CREATE TABLE wp_novarax_tenant_modules (
  id BIGINT(20) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
  tenant_id BIGINT(20) UNSIGNED NOT NULL,
  module_id BIGINT(20) UNSIGNED NOT NULL,
  subscription_id BIGINT(20), -- WooCommerce subscription ID
  status ENUM('active', 'inactive', 'expired', 'cancelled') DEFAULT 'active',
  activated_at DATETIME,
  expires_at DATETIME,
  last_checked DATETIME,
  grace_period_ends DATETIME, -- Allow 7 days grace period
  usage_data LONGTEXT, -- JSON for analytics

  UNIQUE KEY unique_tenant_module (tenant_id, module_id),
  INDEX idx_tenant (tenant_id),
  INDEX idx_module (module_id),
  INDEX idx_subscription (subscription_id),
  INDEX idx_status (status),
  FOREIGN KEY (tenant_id) REFERENCES wp_novarax_tenants(id) ON DELETE CASCADE,
  FOREIGN KEY (module_id) REFERENCES wp_novarax_modules(id) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

```

wp_novarax_api_keys

```

sql

CREATE TABLE wp_novarax_api_keys (
  id BIGINT(20) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
  tenant_id BIGINT(20) UNSIGNED NOT NULL,
  api_key VARCHAR(255) NOT NULL UNIQUE,
  secret_hash VARCHAR(255) NOT NULL,
  permissions TEXT, -- JSON array of allowed endpoints
  rate_limit INT DEFAULT 1000, -- requests per hour
  last_used DATETIME,
  expires_at DATETIME,
  status ENUM('active', 'revoked') DEFAULT 'active',
  created_at DATETIME DEFAULT CURRENT_TIMESTAMP,

  INDEX idx_tenant (tenant_id),
  INDEX idx_api_key (api_key),
  INDEX idx_status (status),
  FOREIGN KEY (tenant_id) REFERENCES wp_novarax_tenants(id) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

```

wp_novarax_audit_logs

sql

```
CREATE TABLE wp_novarax_audit_logs (  
  id BIGINT(20) UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
  tenant_id BIGINT(20) UNSIGNED,  
  user_id BIGINT(20) UNSIGNED,  
  action VARCHAR(255) NOT NULL,  
  entity_type VARCHAR(100), -- 'tenant', 'module', 'user', etc.  
  entity_id BIGINT(20),  
  ip_address VARCHAR(45),  
  user_agent TEXT,  
  details LONGTEXT, -- JSON for additional context  
  created_at DATETIME DEFAULT CURRENT_TIMESTAMP,  
  
  INDEX idx_tenant (tenant_id),  
  INDEX idx_user (user_id),  
  INDEX idx_action (action),  
  INDEX idx_created (created_at)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

wp_novarax_usage_stats

sql

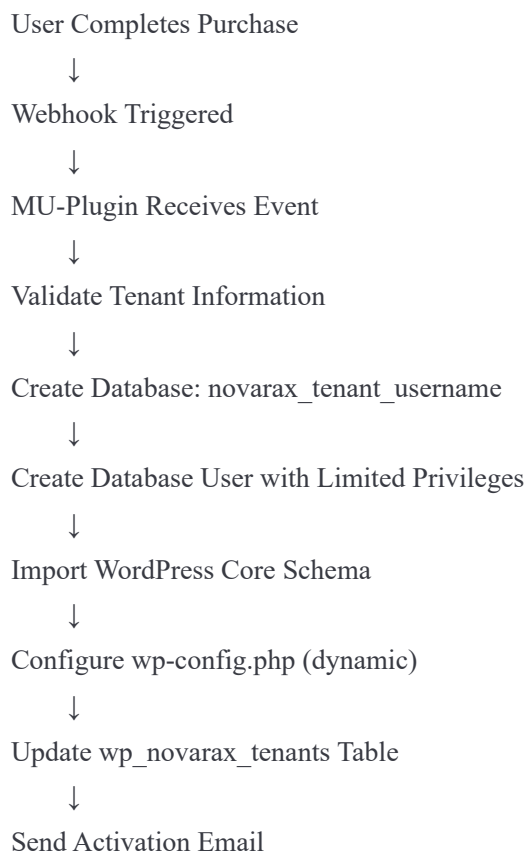
```
CREATE TABLE wp_novarax_usage_stats (  
  id BIGINT(20) UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
  tenant_id BIGINT(20) UNSIGNED NOT NULL,  
  module_id BIGINT(20) UNSIGNED,  
  stat_date DATE NOT NULL,  
  active_users INT DEFAULT 0,  
  api_calls INT DEFAULT 0,  
  storage_used BIGINT(20) DEFAULT 0,  
  page_views INT DEFAULT 0,  
  custom_metrics LONGTEXT, -- JSON for module-specific stats  
  
  UNIQUE KEY unique_tenant_module_date (tenant_id, module_id, stat_date),  
  INDEX idx_tenant (tenant_id),  
  INDEX idx_date (stat_date),  
  FOREIGN KEY (tenant_id) REFERENCES wp_novarax_tenants(id) ON DELETE CASCADE  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

5.3 Database Isolation Strategy

Benefits of Isolated Databases

1. **Security:** Complete data separation between tenants
2. **Compliance:** Easier to meet GDPR, HIPAA requirements
3. **Performance:** No noisy neighbor issues
4. **Backup/Restore:** Granular control per tenant
5. **Customization:** Tenant-specific schema modifications if needed

Database Creation Flow



5.4 Database User Permissions

Master Database User

- Full privileges on `novarax_master`
- No access to tenant databases

Tenant Database Users

sql

```
CREATE USER 'novarax_tenant_username'@'localhost'  
IDENTIFIED BY 'secure_random_password';
```

```
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, INDEX, ALTER  
ON novarax_tenant_username.*  
TO 'novarax_tenant_username'@'localhost';
```

```
FLUSH PRIVILEGES;
```

5.5 Data Migration & Portability

Export Tenant Data

- Admin can export full tenant database as SQL dump
- Includes all module data
- Automated via admin panel or CLI

Import Tenant Data

- Restore from backup
- Migrate from external system
- Merge tenants (advanced feature)

6. Security & Authentication

6.1 Authentication Architecture

Single Sign-On (SSO) Flow

User logs in at app.novarax.ae

- Username/Email + Password
- WordPress validates credentials
- Session created in Master DB

↓

Set Authentication Cookie

- Domain: .app.novarax.ae (wildcard)
- Secure: true (HTTPS only)
- HttpOnly: true (prevent XSS)
- SameSite: Lax
- JWT Token embedded in cookie

↓

User navigates to username.app.novarax.ae

- Browser sends cookie automatically
- Tenant site reads cookie

↓

Tenant MU-Plugin validates token

- Verify JWT signature
- Check expiration
- Query Master API: /api/validate-session
- Confirm user owns this tenant

↓

Create Local Session

- User authenticated on tenant site
- No need to re-enter credentials

JWT Token Structure

json

```
{
  "iss": "app.novarax.ae",
  "sub": "123", // User ID
  "tenant_id": "456",
  "username": "john_doe",
  "email": "john@example.com",
  "iat": 1698753600,
  "exp": 1698840000, // 24 hour expiration
  "tenant_access": ["tenant1", "tenant2"]
}
```

6.2 Security Measures

Password Security

- **Hashing Algorithm:** Bcrypt with cost factor 12
- **Minimum Requirements:**
 - 12 characters minimum
 - Mix of uppercase, lowercase, numbers, symbols
 - No common passwords (dictionary check)
 - No personal information (name, email)
- **Password Reset:** Time-limited tokens (1 hour expiration)
- **Failed Login Protection:** Account lockout after 5 attempts (15-minute cooldown)

Two-Factor Authentication (2FA) - Phase 2

- **Methods:** TOTP (Google Authenticator, Authy)
- **Backup Codes:** 10 single-use codes
- **Enforcement:** Optional for tenants, mandatory for admins

API Security

- **Authentication:** API Keys + Secret Hash
- **Rate Limiting:** 1000 requests/hour per tenant
- **IP Whitelisting:** Optional per tenant
- **Request Signing:** HMAC-SHA256 for sensitive operations

Data Encryption

- **In Transit:** TLS 1.3 (all connections)
- **At Rest:** MySQL transparent data encryption
- **Sensitive Fields:** AES-256 encryption for:
 - API keys
 - Payment information (tokenized)
 - Personal identification data

Access Control

Role	Permissions
Super Admin	Full system access, tenant management, billing, server config
Admin	Tenant CRUD, support tickets, reporting (no server access)
Tenant Owner	Full access to own tenant, subscription management
Tenant Admin	Module management, user management within tenant
Tenant User	Access to assigned modules only
Customer	WooCommerce customer (before provisioning)

Security Monitoring

- **Audit Logging:** All administrative actions
- **Failed Login Tracking:** Real-time alerts on brute force attempts
- **Intrusion Detection:** ModSecurity WAF rules
- **Vulnerability Scanning:** Weekly automated scans
- **Dependency Updates:** Monthly security patches

Compliance Readiness

GDPR Compliance:

- Data portability (export tenant data)
- Right to erasure (delete tenant account + data)
- Data processing agreements
- Cookie consent management

- Privacy policy generator

HIPAA Considerations (if applicable):

- Business Associate Agreements (BAA)
- Encrypted backups
- Audit trails
- Access controls
- Incident response plan

6.3 CORS & Cross-Domain Configuration

php

// Allow authenticated requests between domains

`header('Access-Control-Allow-Origin: https://app.novarax.ae');`

`header('Access-Control-Allow-Credentials: true');`

`header('Access-Control-Allow-Methods: GET, POST, OPTIONS');`

`header('Access-Control-Allow-Headers: Content-Type, Authorization');`

7. Feature Specifications

7.1 Core Platform Features

Tenant Management Dashboard

Admin Features:

1. Tenant List View

- Sortable/filterable table
- Search by name, email, subdomain
- Status indicators (Active, Suspended, Cancelled)
- Quick actions (Edit, Suspend, Delete, Login As)

2. Create New Tenant

- Form validation
- Username availability check (real-time)
- Subdomain preview

- Auto-generate secure passwords
- Email verification

3. Edit Tenant

- Update contact information
- Change subdomain (with migration)
- Adjust storage quotas
- Modify user limits
- Add admin notes

4. Tenant Details Page

- Overview dashboard
- Active subscriptions
- Module usage statistics
- Storage consumption
- User activity logs
- Support ticket history
- Billing information

5. Bulk Operations

- Send announcements
- Suspend multiple tenants
- Export tenant list
- Generate usage reports

Module Marketplace (WooCommerce)

Customer Experience:

1. Browse Modules

- Card-based layout with icons
- Feature highlights
- Pricing information
- Demo videos/screenshots

- Reviews and ratings

2. Module Details Page

- Full description
- Feature list
- System requirements
- Pricing tiers (Monthly/Annual)
- Free trial option (14 days)
- "Add to Cart" button

3. Shopping Cart

- Review selected modules
- Apply discount codes
- Subscription summary
- Estimated monthly/annual cost

4. Checkout Process

- Guest checkout disabled (must be logged in)
- Payment methods:
 - Credit/Debit cards (Stripe)
 - PayPal
 - Bank transfer (for enterprises)
- Billing address
- Terms of service acceptance

5. Post-Purchase

- Order confirmation email
- Provisioning status page
- Estimated completion time (2-5 minutes)
- Redirect to tenant dashboard when ready

Subscription Management

Tenant Features:

1. My Subscriptions Page

- Active modules list
- Renewal dates
- Payment history
- Download invoices

2. Upgrade/Downgrade

- Change billing cycle (monthly ↔ annual)
- Add more modules
- Proration calculations
- Immediate or next-cycle changes

3. Cancellation

- Cancel individual modules
- Cancel entire account
- Grace period (7 days)
- Data export before deletion
- Feedback survey

4. Payment Methods

- Add/remove cards
- Set default payment method
- Update billing address

Admin Features:

1. Subscription Monitoring

- Upcoming renewals
- Failed payments dashboard
- Churn analysis
- Revenue reports

2. Manual Interventions

- Extend trial period

- Apply credits
- Issue refunds
- Pause subscriptions

7.2 Module Specifications

Module 1: Money Manager

Purpose: Personal and business financial tracking

Features:

- **Accounts Management**
 - Multiple account types (Bank, Cash, Credit Card, Investment)
 - Account balances and reconciliation
 - Multi-currency support
- **Transaction Tracking**
 - Income and expense logging
 - Recurring transactions
 - Split transactions
 - Tags and categories
 - Receipt attachments
- **Budgeting**
 - Monthly budget creation
 - Category-based limits
 - Budget vs. actual comparison
 - Alerts for overspending
- **Reporting**
 - Income/Expense reports
 - Cash flow statements
 - Net worth tracking
 - Custom date ranges
 - Export to CSV/PDF

- **Dashboard Widgets**
 - Account summary
 - Recent transactions
 - Spending by category (pie chart)
 - Monthly trends (line chart)

Database Tables:

- wp_mm_accounts
- wp_mm_transactions
- wp_mm_categories
- wp_mm_budgets
- wp_mm_recurring_transactions

Module 2: CRM (Customer Relationship Management)

Purpose: Manage customer relationships and sales pipeline

Features:

- **Contact Management**
 - Contact profiles (name, email, phone, company)
 - Custom fields
 - Contact timeline/activity feed
 - Tags and segments
 - Import/export contacts
- **Company Management**
 - Company profiles
 - Multiple contacts per company
 - Company hierarchy
 - Deal tracking per company
- **Sales Pipeline**
 - Drag-and-drop deal stages
 - Deal value tracking

- Probability estimation
- Expected close dates
- Win/loss reasons
- **Activity Tracking**
 - Calls, emails, meetings
 - Task assignment
 - Follow-up reminders
 - Notes and attachments
- **Reporting**
 - Sales forecasting
 - Conversion rates
 - Sales rep performance
 - Pipeline velocity
 - Revenue reports
- **Email Integration (Phase 2)**
 - Send/receive emails from CRM
 - Email templates
 - Email tracking (opens, clicks)

Database Tables:

- wp_crm_contacts
- wp_crm_companies
- wp_crm_deals
- wp_crm_activities
- wp_crm_pipelines
- wp_crm_tags

Module 3: AI Assistant

Purpose: AI-powered business assistant for various tasks

Features:

- **Conversational Interface**
 - Chat-based UI
 - Context-aware responses
 - Multi-turn conversations
 - Conversation history
- **Use Cases**
 - Content generation (emails, reports, social posts)
 - Data analysis and insights
 - Document summarization
 - Code snippet generation
 - Language translation
 - Customer support responses
- **Integrations**
 - Access CRM data (with permission)
 - Access Money Manager data
 - Generate reports from other modules
 - API access for custom integrations
- **Prompt Library**
 - Pre-built prompts for common tasks
 - Custom prompt templates
 - Share prompts with team
- **Usage Analytics**
 - Tokens consumed
 - Monthly usage limits
 - Cost tracking
 - Popular prompts

Technology Stack:

- OpenAI API (GPT-4)
- Anthropic Claude (alternative)
- Usage-based billing add-on

Database Tables:

- wp_ai_conversations
- wp_ai_messages
- wp_ai_prompts
- wp_ai_usage_logs

Module 4: Loan Management System

Purpose: Manage loan applications, approvals, and repayments

Features:

- **Loan Application**
 - Application forms
 - Document upload
 - Credit scoring integration
 - Applicant profiles
- **Loan Processing**
 - Workflow stages (Applied → Under Review → Approved → Disbursed)
 - Approval/rejection with notes
 - Loan terms configuration (interest rate, tenure)
 - Amortization schedule generation
- **Repayment Management**
 - Payment schedules
 - Payment tracking
 - Late payment alerts
 - Penalty calculation
 - Partial payments

- **Reporting**
 - Outstanding loans
 - Payment collection rates
 - Delinquency reports
 - Loan portfolio analysis
- **Notifications**
 - Payment reminders (email/SMS)
 - Overdue alerts
 - Loan approval notifications
 - Disbursement confirmations

Database Tables:

- wp_loan_applications
- wp_loan_accounts
- wp_loan_payments
- wp_loan_schedules
- wp_loan_documents

7.3 White-Label Features

Admin Dashboard Customization

- Custom logo upload
- Color scheme customization
- Custom login page
- Remove "WordPress" branding
- Custom admin menu structure
- Custom footer text

Tenant Dashboard Customization

- Tenant-specific branding
- Custom domain support (Phase 2: custom.clientdomain.com)

- Email template customization
- Custom CSS injection
- White-label mobile app (Phase 3)

Recommended White-Label Plugins

1. **White Label CMS** - Hide WordPress branding
2. **Admin Menu Editor** - Customize menu structure
3. **Login Designer** - Custom login pages
4. **Custom Admin Dashboard** - Replace default dashboard

7.4 Reporting & Analytics

Admin Analytics Dashboard

Key Metrics:

- Total active tenants
- Monthly Recurring Revenue (MRR)
- Annual Recurring Revenue (ARR)
- Customer Acquisition Cost (CAC)
- Customer Lifetime Value (LTV)
- Churn rate
- Module adoption rates
- Average revenue per user (ARPU)

Charts & Visualizations:

- Revenue trend (line chart)
- New tenants vs. churned (bar chart)
- Module popularity (pie chart)
- Geographic distribution (map)
- Subscription tier distribution

Reports:

- Daily sales summary

- Monthly financial report
- Tenant activity report
- Support ticket metrics
- Server resource usage

Tenant Analytics Dashboard

Available to Tenants:

- Module usage statistics
- Storage consumption
- User activity (logins, actions)
- API usage (if applicable)
- Feature adoption rates

7.5 Support & Communication

Support Ticket System

Features:

- Ticket submission form
- Priority levels (Low, Medium, High, Urgent)
- Category tagging (Technical, Billing, Feature Request)
- File attachments
- Email notifications
- Ticket status tracking (Open, In Progress, Resolved, Closed)
- SLA tracking for response times

Admin Features:

- Ticket queue management
- Assign to team members
- Internal notes
- Canned responses
- Ticket merging

- Knowledge base integration

Communication Channels

- In-app messaging
- Email notifications (transactional)
- System announcements
- Newsletter integration
- SMS notifications (Phase 2)

Knowledge Base

- Getting started guides
 - Module documentation
 - Video tutorials
 - FAQ section
 - API documentation
 - Troubleshooting guides
-

8. Development Phases

Phase 1: Foundation & Infrastructure (Weeks 1-4)

Week 1: Environment Setup

- Provision IONOS VPS server
- Install Ubuntu 22.04 LTS
- Configure firewall (UFW)
- Install and configure:
 - Nginx web server
 - MySQL 8.0 database server
 - PHP 8.2 with required extensions
 - Redis cache server
- Set up wildcard DNS records

- Install wildcard SSL certificate
- Configure automated backups

Deliverables:

- Server fully operational
- All three WordPress installations running
- SSL certificates active
- Documentation: Server setup guide

Week 2: WordPress Base Installation

- Install WordPress on:
 - novarax.ae (marketing site)
 - app.novarax.ae (admin dashboard)
 - /var/www/tenant-dashboard (shared codebase)
- Install WooCommerce + WooCommerce Subscriptions
- Install white-label plugins
- Configure permalink structure
- Set up Redis object caching
- Create initial admin accounts

Deliverables:

- Three WordPress instances configured
- WooCommerce functional
- Basic theme customization
- Documentation: WordPress setup guide

Week 3: Database Architecture

- Design and create custom tables
- Write database migration scripts
- Implement database user management
- Set up automated database backup system

- Create database documentation
- Performance optimization (indexes, query optimization)

Deliverables:

- All custom tables created
- Migration scripts tested
- Backup system operational
- Documentation: Database schema reference

Week 4: Version Control & Deployment

- Set up Git repositories
- Configure staging environment
- Create deployment scripts
- Document Git workflow
- Set up continuous integration (CI)
- Implement code review process

Deliverables:

- Git repositories configured
 - Staging environment live
 - Deployment process documented
 - Documentation: Development workflow guide
-

Phase 2: Core Tenant Management (Weeks 5-8)**Week 5: Tenant Manager MU-Plugin (Part 1)**

- Create plugin structure
- Build tenant registration system
- Implement username validation
- Create tenant database provisioning logic
- Build subdomain routing handler

- Develop tenant status management

Deliverables:

- Tenant creation functional
- Database auto-provisioning working
- Subdomain routing operational
- Documentation: Tenant Manager API

Week 6: Tenant Manager MU-Plugin (Part 2)

- Build admin UI for tenant management
- Create tenant list view with filters
- Implement tenant edit functionality
- Build tenant details dashboard
- Add bulk operations
- Create tenant deletion with cleanup

Deliverables:

- Complete admin interface
- All CRUD operations functional
- Tenant dashboard displaying metrics
- Documentation: Admin user guide

Week 7: WooCommerce Integration

- Create product-to-plugin mapping system
- Build subscription webhook handlers
- Implement post-purchase automation
- Develop provisioning queue system
- Create email notification templates
- Test purchase-to-provisioning flow

Deliverables:

- WooCommerce fully integrated
- Purchase triggers provisioning
- Email notifications working
- Documentation: WooCommerce integration guide

Week 8: Testing & Refinement

- End-to-end testing (registration → purchase → provisioning)
- Load testing (simulate 50 concurrent provisioning)
- Fix bugs and edge cases
- Performance optimization
- Security audit
- Documentation review

Deliverables:

- Bug-free tenant management system
 - Performance benchmarks met
 - Security vulnerabilities addressed
 - Documentation: Testing report
-

Phase 3: Authentication & SSO (Weeks 9-11)

Week 9: SSO Implementation

- Implement JWT token generation
- Configure shared cookie domain
- Build token validation system
- Create SSO API endpoints
- Implement session synchronization
- Add logout across domains

Deliverables:

- SSO functional between domains

- Users authenticated seamlessly
- Secure token management
- Documentation: SSO architecture guide

Week 10: Security Hardening

- Implement rate limiting
- Add brute force protection
- Configure CORS properly
- Add CSRF protection
- Implement XSS prevention
- Set up security headers

Deliverables:

- Security measures implemented
- Penetration testing passed
- Security audit report
- Documentation: Security best practices

Week 11: User Management

- Build user roles and capabilities
- Create tenant user management UI
- Implement team member invitations
- Add user activity logging
- Create user permission system
- Build profile management

Deliverables:

- Complete user management system
- Role-based access control working
- Team collaboration enabled
- Documentation: User management guide

Phase 4: Module License System (Weeks 12-14)

Week 12: License Server (Admin Side)

- Build module registration system
- Create product-to-module mapping UI
- Implement subscription status API
- Build license validation endpoints
- Create module activation hooks
- Develop usage tracking system

Deliverables:

- License server operational
- API endpoints documented
- Module registration UI complete
- Documentation: License system architecture

Week 13: License Client (Tenant Side)

- Build tenant-side license checker
- Implement plugin activation filter
- Create grace period logic
- Build subscription expiry notifications
- Add "renew subscription" prompts
- Implement offline mode (24hr cache)

Deliverables:

- Dynamic plugin activation working
- Expired subscriptions handled gracefully
- User notifications functional
- Documentation: License client guide

Week 14: Module Marketplace UI

- Design and build marketplace page
- Create module cards with details
- Implement shopping cart
- Build checkout flow
- Add trial period logic
- Create "my subscriptions" page

Deliverables:

- Polished marketplace interface
 - Smooth checkout experience
 - Trial periods functional
 - Documentation: Marketplace user guide
-

Phase 5: First Module - Money Manager (Weeks 15-18)

Week 15-16: Core Development

- Design database schema
- Build backend API (REST endpoints)
- Develop account management
- Create transaction system
- Implement category management
- Build recurring transactions

Deliverables:

- Backend fully functional
- API documented
- Database optimized
- Documentation: Money Manager API reference

Week 17: Frontend Development

- Design UI/UX
- Build React/Vue components (or WordPress Gutenberg)
- Create dashboards and charts
- Implement transaction forms
- Build reporting interface
- Mobile-responsive design

Deliverables:

- Complete user interface
- Interactive dashboards
- Beautiful visualizations
- Documentation: Money Manager user manual

Week 18: Testing & Polish

- Unit testing
- Integration testing
- User acceptance testing (UAT)
- Bug fixes
- Performance optimization
- Documentation finalization

Deliverables:

- Production-ready module
- Test coverage report
- Performance benchmarks
- Documentation: Money Manager complete guide

Phase 6: Additional Modules (Weeks 19-26)

Week 19-21: CRM Module

- Database design

- Contact management system
- Company management
- Deal pipeline
- Activity tracking
- Reporting dashboards

Deliverables:

- Functional CRM module
- Complete documentation

Week 22-24: AI Assistant Module

- OpenAI API integration
- Conversation interface
- Prompt library
- Usage tracking
- Token management
- Cost calculation

Deliverables:

- Working AI assistant
- Integration with other modules
- Complete documentation

Week 25-26: Loan Management Module

- Application workflow
- Loan processing
- Payment schedules
- Amortization calculations
- Reporting system
- Document management

Deliverables:

- Complete loan management system
 - Financial calculations accurate
 - Complete documentation
-

Phase 7: White-Labeling & Customization (Weeks 27-28)**Week 27: Branding System**

- Implement tenant branding options
- Create logo upload system
- Build color customization
- Design custom login pages
- Remove all WordPress branding
- Create email template system

Deliverables:

- Complete white-label system
- Tenant customization options
- Professional appearance
- Documentation: Branding guide

Week 28: Theme Development

- Custom admin theme for master dashboard
- Custom tenant theme
- Responsive design
- Accessibility compliance (WCAG 2.1)
- Performance optimization
- Cross-browser testing

Deliverables:

- Professional custom themes

- Mobile-optimized
 - Accessibility certified
 - Documentation: Theme customization guide
-

Phase 8: Testing, Optimization & Launch (Weeks 29-32)

Week 29: Comprehensive Testing

- System integration testing
- Load testing (100+ concurrent users)
- Security penetration testing
- Browser compatibility testing
- Mobile device testing
- Accessibility audit

Deliverables:

- Complete test report
- All critical bugs fixed
- Performance benchmarks met
- Documentation: QA report

Week 30: Performance Optimization

- Database query optimization
- Caching strategy implementation
- CDN configuration
- Image optimization
- Minification and compression
- Lazy loading implementation

Deliverables:

- Page load times < 3 seconds
- Server response times < 500ms

- Optimized resource usage
- Documentation: Performance tuning guide

Week 31: Documentation & Training

- Complete admin documentation
- User manuals for all modules
- Video tutorials
- API documentation
- Troubleshooting guides
- Team training sessions

Deliverables:

- Comprehensive documentation portal
- Training videos
- Knowledge base articles
- Documentation: Complete platform guide

Week 32: Soft Launch & Monitoring

- Deploy to production
- Set up monitoring tools (Uptime, APM)
- Configure error logging
- Implement analytics
- Invite beta users (10-20 tenants)
- Collect feedback and iterate

Deliverables:

- Platform live in production
 - Monitoring dashboards active
 - Beta user feedback collected
 - Documentation: Launch checklist
-

9. Team Structure & Responsibilities

9.1 Recommended Team Composition

