Project 7

Ray Biner Real Estate Management System

About the client:

Ray Biner Real Estate commenced serving the suburbs of Gladstone Park, Tullamarine, Westmeadows, Attwood, Greenvale, Mickleham, Yuroke and Bulla some 25 years ago and has gone from strength to strength while building a reputation unsurpassed.

We are the longest serving estate agency in the area and our hand-picked Sales and Property Management team have extensive collective experience in all facets of real estate.

We are all fully trained professionals with knowledge of real estate rules, regulations and financing requirements. We have professional Auctioneers within our organisation that are fully trained and very competent in obtaining the maximum value for your property.

Our distinctive media advertising, website, digitally enhanced office front and signboards all contribute towards a successful brand to which our clients prefer to align themselves. Our dynamic Sales Team have won Top Auction, Top Sales Executive and Top Franchise for 4 consecutive years and established itself as Top Office in Victoria.

Proposed System Overview

This system enables property owners, agents, and clients to manage, buy, sell, or rent real estate properties online. The system provides separate dashboards for administrators, property owners/agents, and buyers/tenants, supporting property listings, user management, payments, and communication through an intuitive web interface.

System Modules

- User Management
- Property Management
- Property Search & Filtering
- Property Viewing & Enquiry
- Booking & Scheduling
- Payment & Transaction
- Communication & Notification
- Reports & Analytics
- Admin Control Panel
- AI & Automation

Functional Requirements

Functional requirements are product features or functions that developers must implement to enable users to accomplish their tasks. So, it's important to make them clear for the stakeholders. Generally, functional requirements describe system behavior under specific conditions. The developers of this system must enhance the performance and efficiency of the system by adding 15 to 20 more functional requirements. Students need to do their own research to find how they can improve the system and which FRs need to added. The group must need a prior approval from the stakeholders/project supervisor before finalizing these Functional Requirements. These enhanced FRs must be reflected separately in Final SRS Report after the approval.

User Registration and Authentication

FR1. The system should allow new users (Admin, Property Owner/Agent, Buyer/Tenant) to register with basic details including name, email, password and phone number.
FR2. The system should send an email verification link to validate user registration.
FR3. The system should allow users to log in securely using verified credentials.
FR4. The system should allow users to reset passwords through an email recovery option.
FR5. The system should maintain different roles and permissions for Admin,

Property Owner/Agent and Buyer/Tenant.
FR6. The system should allow Admins to suspend or reactivate user accounts.

User Profile Management

FR7. The system should allow users to update personal information (name, phone, profile picture, etc.)
FR8. The system should allow Property Owners/Agents to add company details, licenses or certifications.
FR9. The system should allow Buyers/Tenants to maintain a wishlist or saved property list.

Property Management

FR10. The system should allow Property Owners/Agents to add new property listings with details (title, type, location, price, description).
FR11. The system should allow uploading of multiple property images or videos.
FR12. The system should allow Owners to update or edit property information.
FR13. The system should allow Owners to remove or deactivate listings.
FR14. The system should allow Admin to approve or reject property listings before publishing.
FR15. The system should display status tags such as Available, Rented, Sold or Pending.
FR16. The system should allow Admins to assign categories and locations to properties.
FR17. The system should maintain a property ID and ownership history for each property.

Property Search and Filtering

FR18. The system should allow users to search properties using keywords (e.g. city, type, price).
FR19. The system should provide advanced filters (price range, size, bedrooms, property type).
FR20. The system should allow sorting results by price, date listed or popularity.

FR21. The system should display property listings in grid and list views.
FR22. The system should allow map-based property browsing using Google Maps integration.

Property Details and Viewing

FR23. The system should display a detailed property page with full information, images and owner details.
FR24. The system should show similar property recommendations based on price range and location.
FR25. The system should allow registered users to contact property owners/agents via an inquiry form.
FR26. The system should record all inquiries in the database for future reference.

Booking and Scheduling

FR27. The system should allow Buyers/Tenants to book property viewings online.
FR28. The system should notify Owners/Agents of new booking requests via dashboard and email.
FR29. The system should allow Owners to accept or reject booking requests.
FR30. The system should automatically update the property's status after a confirmed booking.

Payment and Transaction Management

FR31. The system should allow Buyers/Tenants to make payments or deposits through a secure gateway (e.g. PayPal Sandbox).
FR32. The system should generate digital receipts or invoices for all successful transactions.
FR33. The system should allow Admins to review, verify and approve transactions.
FR34. The system should allow Owners to view transaction history and pending payments.

Communication and Messaging

FR35. The system should include a messaging module for direct communication between users.
FR36. The system should send email or system notifications for inquiries, approvals and payments.
FR37. The system should allow Admins to broadcast announcements to all users.

Reports and Analytics

FR38. The system should allow Admins to generate reports on property listings, sales and rentals.
FR39. The system should provide visual analytics for user activity and traffic trends.
FR40. The system should allow data export (CSV or PDF) for reporting and audit purposes.

Security and Data Management

FR41. The system should validate and sanitise all inputs to prevent SQL injection or XSS.
FR42. The system should store passwords securely using hashing (bcrypt or MD5).
FR43. The system should allow Admins to view and manage system logs for all transactions and updates.

AI and Automation Features

FR44. The system should use a basic AI recommendation engine to suggest similar properties based on search history and preferences.
FR45. The system should include a chatbot or virtual assistant that responds to frequently asked questions (FAQs) about listings, registration or payments.

Non-Functional Requirements

Performance

NFR1: System pages should load within 3 seconds under normal network conditions.

NFR2: The system should handle at least 100 concurrent users without performance degradation.

NFR3: Database queries should execute within 1 second for typical transactions.

Usability

NFR4: The user interface should be intuitive and easy to navigate for non-technical users.

NFR5: The system should be responsive and mobile-friendly, viewable on desktops, tablets and smartphones.

NFR6: All forms and actions should provide clear feedback messages (success, error, or validation).

Reliability & Availability

NFR7: The system should achieve 99% uptime excluding scheduled maintenance.

NFR8: The database should be backed up automatically every 24 hours.

NFR9: In the event of a crash the system should recover data from the last successful backup.

Security

NFR10: All passwords should be stored using secure hashing algorithms (e.g. bcrypt).

NFR11: The system should enforce HTTPS for all client-server communications.

NFR12: The system should include input validation and sanitisation to prevent SQL injection and XSS attacks.

NFR13: User sessions should auto-expire after 10 minutes of inactivity.

Maintainability

NFR14: The system should be implemented using the MVC (Model-View-Controller) structure for modularity.

NFR15: The codebase should include comments and documentation for future developers.

Scalability

NFR16: The database design should support future expansion (e.g. adding mortgage or maintenance modules).

Compatibility

NFR17: The system should be compatible with PHP ≥ 8.0 and MySQL ≥ 8.0.

NFR18: The system should support modern browsers (Chrome, Firefox, Safari and Edge).

Hardware Requirement: Should be recommended by the developers.
Software Requirement: Should be recommended by the developers.