iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** VIII **Month of publication:** August 2025

DOI: https://doi.org/10.22214/ijraset.2025.73851

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VIII Aug 2025- Available at www.ijraset.com

Web-Based Real Estate Management System: DBMS-Supported Solution for Property Listings and Buyer-Seller Interaction

Sirisha Paidi[1], Pranay Vemula[2]
[1]*Student, School of Informatics, Department of MCA [Data Science], Aurora Deemed University, Hyderabad*
[2]*Assistant Professor, School of Engineering, Department of CSE, Aurora Deemed University, Hyderabad*

Abstract: ***This paper introduces the design and development of a web-based Real Estate Management System (REMS) that facilitates hassle-free interactions between buyers and sellers of property via an efficient and secure platform. Conventional real estate transactions, which are usually based on offline procedures or isolated online postings, are marred by issues like deficiency in transparency, inefficiency, and limited access. The system proposed herein solves these problems by offering a central, database-based property registration, listing management, and buyer-seller communication solution. The backend is assisted with a Database Management System (DBMS) for the secure storage, efficient retrieval, and scalable management of property information and user accounts. The frontend employs an easy-to-use interface, facilitating the registration of sellers and updating of property information as well as the searching, filtering, and browsing of properties by location, budget, and size preference for buyers. The system also has additional features such as authentication techniques to secure sensitive user information and direct contact capabilities for ease of communication. Through the incorporation of database security, responsive layouts, and real-time access to listings, the system increases transparency, efficiency, and trust in property dealings. This project is helping achieve the digitalization of the real estate industry through offering an affordable and scalable platform for contemporary real estate management requirements.***

Keywords: Real Estate Management, Web Application, DBMS, Property Listings, Buyer-Seller Communication, User Interface, Secure Transactions

I. INTRODUCTION

Correct handling of property transactions is a critical need in the current real estate sector. Conventional processes of selling and purchasing real estate generally depend on offline communication, print media advertisements, or disjointed online listings that are time-consuming, opaque, and that offer restricted access to sound information. These shortcomings are barriers for property buyers and also sellers alike and lead to late decision-making, ineffective communication, and decreased trust in the process as a whole.

The development of web technologies and database applications has unlocked spaces to develop centralized platforms with accessibility, security, and scalability. A Web-Based Real Estate Management System (REMS) fills in the gaps of the traditional approaches by having one platform where the property owners can post, update, and manage their properties and the buyers can search, filter, and view the properties that suit their needs best in real-time. Such a system not only improves efficiency but also ensures transparency, accessibility, and trustworthiness in property transactions.

The suggested system makes use of a Database Management System (DBMS) as the backend to safely store and handle property information, user data, and transaction data. The user interface has been made intuitive and user-focused so that the sellers can create and update listings of properties and buyers can search according to location, price, and size filters. Other than this, the site features authentication measures for secure protection of sensitive data and offers direct communication interfaces between the buyer and seller to minimize reliance on middlemen.

In addition to functionality, the system is based on transparency and trust through the use of features like rating and feedback systems so that buyers and sellers can make well-informed decisions. With secure database integration, responsive web design, and dynamic web technologies, the system is a step closer to automating the real estate sector.

This paper presents the architecture, design, and development of the Web-Based Real Estate Management System with the purpose of solving real issues that are experienced in property transactions and according to prevailing software engineering best practices.

©IJRASET: All Rights are Reserved

II. LITERATURE REVIEW

Real estate information systems have progressed from offline broker networks and desktop databases to web-based, database-backed platforms that support large catalogs, rich search, and multi-party interactions. Prior work spans listing portals, GIS-enabled search, recommender systems, and secure, role-based marketplaces.

Despite progress, common gaps persist: data authenticity, fragmented buyer-seller communication, limited transparency, and inconsistent security practices.

Singh and Mehta [1] proposed a Web-Based Real Estate Listing and Management System centralizing property records with faceted search (location, size, budget). While improving discovery and basic CRUD operations, the platform lacked verified seller profiles, secure in-platform messaging, and audit trails for edits, limiting transaction trust.

Li and Zhao [2] introduced preference-aware property matching, modeling buyer constraints to rank listings. Their approach improved relevance but provided minimal privacy controls and no explainability for ranking decisions, constraining user trust and adoption.

Ahmed et al. [3] reviewed large commercial portals, noting strengths in scale and market liquidity but recurring issues in data duplication, stale listings, and intermediary dependency. They recommended standardized data models and direct communication channels to reduce friction.

Rao and Gupta [4] implemented a GIS-enabled Real Estate Information System integrating map-based search (e.g., buffer queries, proximity to amenities) atop a relational DBMS. Although spatial indexing improved user experience, the system faced scalability bottlenecks and lacked role-based access and privacy-preserving location handling.

Fernandes and Costa [5] explored secure buyer–seller communication within property portals via platform messaging and escrow-like workflows. Security improved versus email/phone exchanges, but the solution omitted end-to-end encryption, KYC/verification workflows, and reputation mechanisms.

Patel and Kumar [6] examined DBMS schema design for property catalogs, contrasting normalized RDBMS designs with schema-flexible NoSQL for fast ingestion and variable attributes (amenities, rules). They emphasized indexing strategies (composite, text, geo) but did not address analytics pipelines (e.g., search logs, conversion funnels).

Nielsen [7] outlined usability heuristics for marketplace UIs, highlighting discoverability, feedback, and error prevention. Studies showed that poor filter design, ambiguous labels, and hidden actions reduce engagement and conversion—issues prevalent in real estate portals with dense forms.

OWASP Foundation [8] documented best practices for authentication and authorization in web apps, recommending JWT with short lifetimes, refresh tokens, and role-based access control. Prior real estate systems often implemented incomplete flows, leading to session fixation risks and weak authorization boundaries.

Fielding [9] articulated RESTful service constraints underpinning scalable APIs. Real estate platforms adopting resource-oriented endpoints, stateless interactions, and HATEOAS improved maintainability, yet many lacked consistent pagination, caching, and error semantics crucial for performance and client developer experience.

PostGIS Documentation [10] described spatial extensions for RDBMS, enabling geo-indexes, distance filters, and polygon containment. Prior systems using PostGIS reported marked speedups for nearby-search but noted maintenance complexity and the need for careful query planning at scale.

Johnson [11] discussed cloud-native deployment (containerization, autoscaling, managed DB) for transaction platforms, improving uptime and cost profiles.

However, multi-tenant isolation, backup/restore drills, and observability (tracing, metrics) were inconsistently adopted in real estate contexts.

Kiran and Rathi [12] surveyed reputation systems (ratings/reviews) to enhance marketplace transparency. While ratings helped reduce adverse selection, they cautioned about sybil attacks, retaliation bias, and the need for moderation workflows and appeals.

Collectively, these works inform the proposed Web-Based Real Estate Management System (REMS). REMS combines (i) a DBMS-backed data layer with text and geo-indexes for fast, precise search; (ii) secure authentication (JWT-based) and role-aware access control; (iii) in-platform messaging to streamline negotiations; and (iv) transparency features (ratings/feedback, change audit logs). By addressing gaps in data authenticity, security, communication, and usability, REMS advances the state of practice toward a trustworthy, efficient, and scalable property transaction platform.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VIII Aug 2025- Available at www.ijraset.com

III.METHODOLOGY

A. Existing Methodology

Traditional Real Estate Management Systems (REMS) primarily relied on manual processes such as paperwork, spreadsheets, and fragmented desktop applications for managing property transactions. These approaches, while sufficient for small-scale operations, often resulted in inefficiencies, human errors, limited accessibility, and lack of transparency. With the emergence of digital technologies, several web-based solutions have been developed to centralize property listings, streamline buyer-seller interactions, and provide administrators with greater control. However, existing methodologies still reveal gaps in scalability, security, interoperability, and user-centric design.

1) Manual and Semi-Digital Real Estate Management:

This approach represents the traditional workflow of real estate managers and property owners using physical records or basic spreadsheet software.

Key characteristics include:

- Paper-based documentation of property ownership and transactions.
- Fragmented use of spreadsheets for rent collection, tenant details, and property listings.
- Offline communication between buyers, sellers, and agents.
- Limited data backup and high risk of loss/damage.

Limitations:

- Highly prone to errors and inconsistencies.
- No centralized data storage, making information retrieval difficult.
- Lack of authentication or secure data handling.
- Time-consuming and inefficient for large-scale property portfolios.

2) Early Web-Based Property Portals

With advances in web technologies, real estate portals were introduced to digitize property listings and basic buyer-seller interactions.

Key characteristics include:

- Centralized property listing databases with search functionality.
- Basic buyer/seller registration and login modules.
- Limited filtering options (e.g., location, price, property type).
- Manual reporting through downloadable property lists in PDF/Excel formats.
- Basic contact information exchange between buyers and sellers.

Drawbacks:

- Limited authentication and weak data privacy measures.
- Lack of direct communication tools (buyers often rely on external channels).
- Minimal support for real-time updates and notifications.
- No advanced recommendation or matching algorithms.
- Absence of transparency features such as ratings, reviews, or audit trails.

3) Real Estate Management in Enterprise Applications

Some enterprise-level solutions integrate real estate management modules into larger ERP or CRM systems.

Key characteristics include:

- Centralized property and tenant databases.
- Role-based access for administrators, agents, and owners.
- Financial modules for rent collection and expense reporting.
- Compliance monitoring with local property regulations.

©IJRASET: All Rights are Reserved

Limitations:

- High deployment and maintenance costs.
- Limited scalability for small property owners or individual sellers.
- Minimal flexibility for end-user customization.
- Often complex, requiring technical expertise to operate.

B. Proposed Methodology

The proposed methodology introduces a Graphical User Interface (GUI)-based Real Estate Management System (REMS UI) that streamlines property listing, buyer search, and administrative management through a centralized, database-driven platform. The system is implemented using Visual Studio as the development environment and SQL Server for backend data storage and management.

1) System Architecture

The system architecture consists of three primary modules, supported by a database-driven backend for secure and efficient data handling.

- Frontend: Developed in Visual Studio, offering a GUI-based interface with dedicated forms for sellers, buyers, and administrators.
- Backend: SQL Server manages structured data storage, retrieval, and updates for property listings and user records.
- Authentication: Basic validation mechanisms to ensure secure registration and access to the platform.
- Database: Centralized repository storing seller details, buyer requirements, and property information.

2) Functional Modules

The system is divided into three main operational modules, each accessed through the main interface:

- Sale a Plot (Seller Module):
  - Sellers register their personal and property details, including name, contact number, address, plot registration number, location, area, price, and previous ownership.
  - Upon submission, data is stored securely in the SQL Server database for later retrieval and display.
- Buy a Plot (Buyer Module):
  - Buyers enter their requirements (e.g., location and area) through combo box selections.
  - The system queries the database and displays matching property listings in a grid view for user convenience.
- Management Module (Admin Module):
  - Enables administrative tasks such as updating, deleting, and searching property records.
  - Updates are performed by entering new details for a record, while deletion and search operations are carried out using the property registration number.

3) User Interface Design

- The first interface provides sellers with structured forms for entering property and ownership details.
- The second interface supports buyers with dynamic search filters (location, area) and displays results in a tabular grid view.
- The third interface provides administrative control for database management operations.

4) Data Management and Security

- SQL Server ensures structured storage of property and user data, allowing efficient retrieval and manipulation.
- Validation mechanisms are integrated within forms to minimize entry errors and maintain data integrity.
- Centralized data handling promotes accuracy and consistency across the platform.

IV. SYSTEM DESIGN AND ARCHITECTURE

The proposed Real Estate Management System (REMS) follows a three-tier architecture comprising the Presentation Layer, Business Logic Layer, and Data Layer. This modular design ensures clarity of operations, user-friendly navigation, and robust handling of real estate transactions.

A. Architectural Overview

Presentation Layer (Frontend – Windows Forms GUI)

- Developed using Visual Studio (C++/CLI Windows Forms), this layer provides the graphical user interface (GUI) through multiple forms:
- MyForm: The main navigation hub offering options to Sell, Buy, or Manage plots.
- MyForm1 (“Sale a Plot”): Data entry form where sellers register plot details (Owner Name, Phone, Address, Plot Registration No., Location, Area, Price, Previous Owner).
- MyForm2 (“Buy a Plot”): Buyer search form with filtering options (Location, Area) that displays matching plots in a grid view.
- MyForm3 (“Management”): Central form for accessing update, delete, and search operations.
- MyForm4 (“Edit Records”): A detailed CRUD interface for modifying existing records, deleting entries, or performing search queries.

Business Logic Layer (Application Layer)

- Each form contains its own event-driven logic that handles data validation, CRUD operations, and database queries:
- Validation Logic (MyForm1): Ensures mandatory fields are filled, prevents duplicate entries, and enforces data consistency.
- Search/Filter Logic (MyForm2): Constructs SQL SELECT queries based on buyer requirements and displays results in a DataGridView.
- Management Logic (MyForm3/MyForm4): Handles UPDATE, DELETE, and advanced search operations using Plot Registration No. as the key.

Data Layer (SQL Server Database)

- A single Microsoft SQL Server database named jsrrealestate stores all property listings.
- Primary Table: management5
- Attributes: OwnerName, Phone, Address, PlotRegNo, PlotLocation, PlotArea, PlotPrice, PreviousOwner.
- Data Access: Implemented using SqlConnection, SqlCommand, SqlDataAdapter, and SqlDataReader objects for reliable interaction with the database.

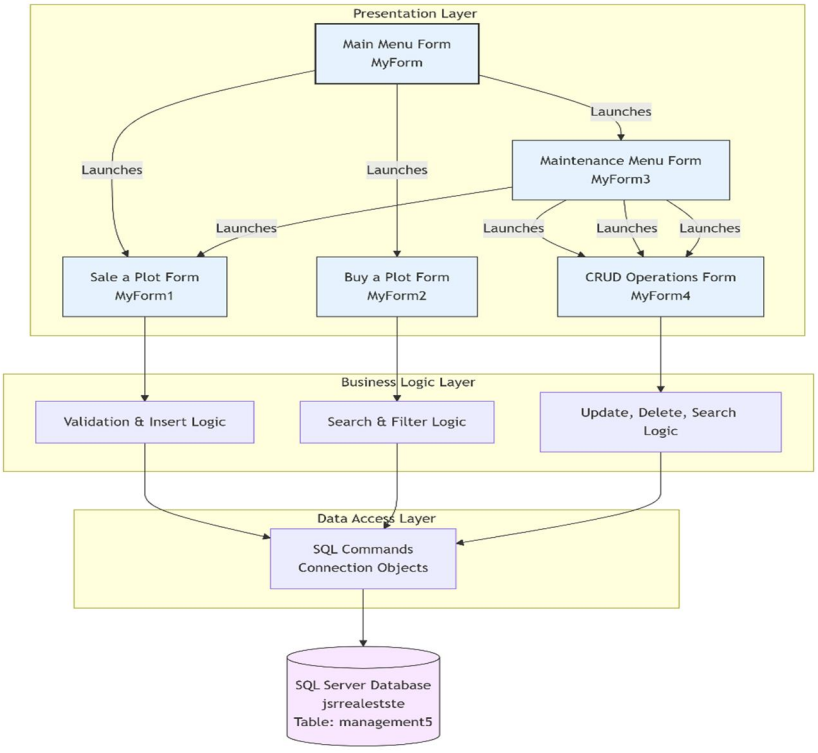


Fig. 1: System Architecture Diagram

B. Security and Scalability Considerations

Security:

- Input validation prevents SQL errors and inconsistent data entry.
- Sensitive identifiers (Plot Registration Numbers) are used for safe update/delete operations.
- Exception handling ensures database errors are properly caught and reported.

Scalability:

- Additional forms/modules (e.g., Lease Management, Payment Tracking) can be easily integrated without altering the existing database structure.
- SQL Server allows scaling to larger datasets with indexing and stored procedures.

Extensibility:

- Features such as advanced search filters (price range, owner history), report generation (PDF/Excel), or integration with GIS/Mapping APIs can be added in future versions.

C. Data Flow Diagram (DFD):

External Entities:

- Real Estate Agent/User: The primary user inputting data, searching for plots, and managing records.

Data Flows:

- Inputs Data & Commands: This includes all user interactions: adding new plot details, entering search criteria (location, area), requesting to update or delete a record, and asking to view all data.
- Stores/Retrieves Data: The system's bi-directional communication with the database to save new records, update existing ones, delete entries, and query for information.
- Displays Results & Messages: The system's output to the user, including success/error messages, search results displayed in DataGridViews, and confirmation dialogs.

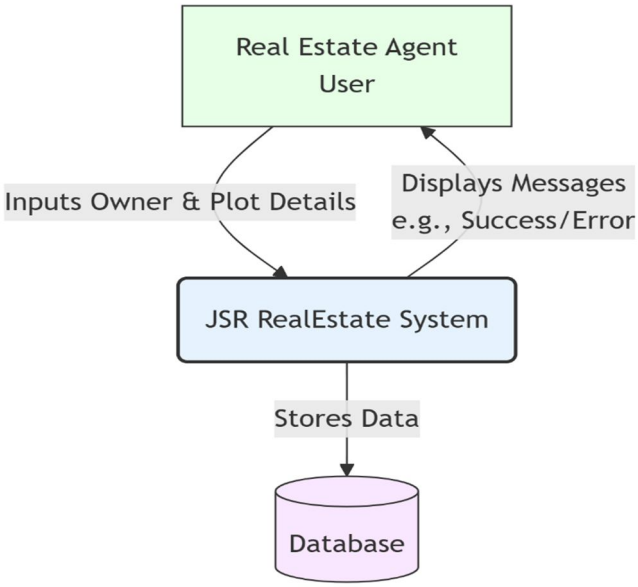


Fig. 2: Data Flow Diagram - Level 0 (Context Diagram)

Processes:

- Add New Plot: Validates input and inserts a new record into the management5 table.
- Search/Buy Plot: Queries the database for plots matching the selected location and area criteria.
- Update Record: Modifies existing records in the database based on the user's input.
- Delete Record: Removes a record from the database, typically using the plot_regno as the key.
- View All Data: Retrieves and displays all records from the management5 table.

Data Store:

- Database: The single data store for all processes.

Data Flows:

- The flows clearly show the type of SQL command each process uses and the nature of the data returned (e.g., query results for searches, simple success/failure messages for updates).

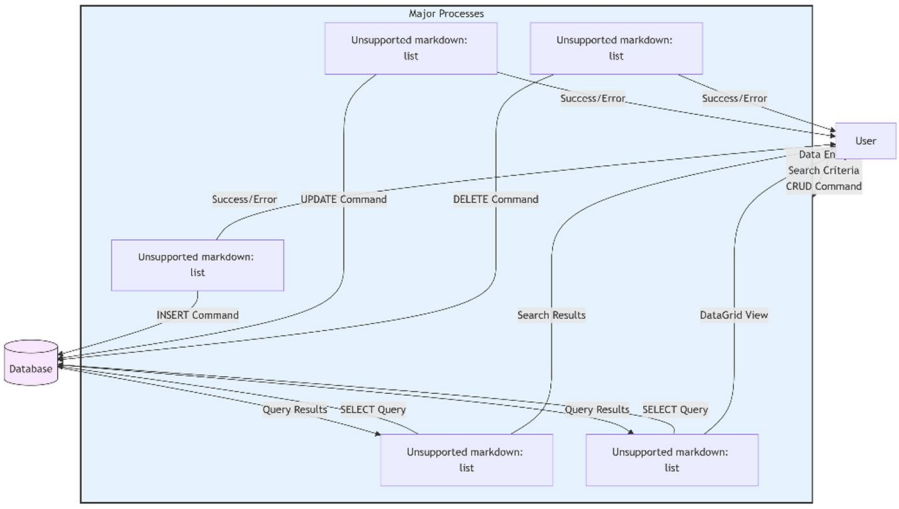


Fig. 3: Data Flow Diagram - Level 1

Interpretation:

- This design ensures owners can easily register plots, buyers can search properties seamlessly, and administrators can manage records efficiently, all backed by a secure SQL Server database.

V. IMPLEMENTATION

The Real Estate Management System (REMS) is implemented using C++/CLI Windows Forms (Visual Studio) for the user interface and Microsoft SQL Server for the database. The system adopts a modular architecture, where each major functionality (e.g., adding a new plot, searching, updating records) is handled through separate forms and event-driven methods.

A. Frontend Implementation

The frontend is developed using Windows Forms GUI in Visual Studio. Each major functionality is mapped to a dedicated form, ensuring modularity and ease of use.
Main Navigation: Provides three entry points through buttons: Sale a Plot, Buy a Plot, and Management.
DataGridView Integration: For displaying search results and records.
Error Handling: MessageBox alerts for invalid or missing data.
User-Friendly Design: Simplifies navigation and reduces training needs.

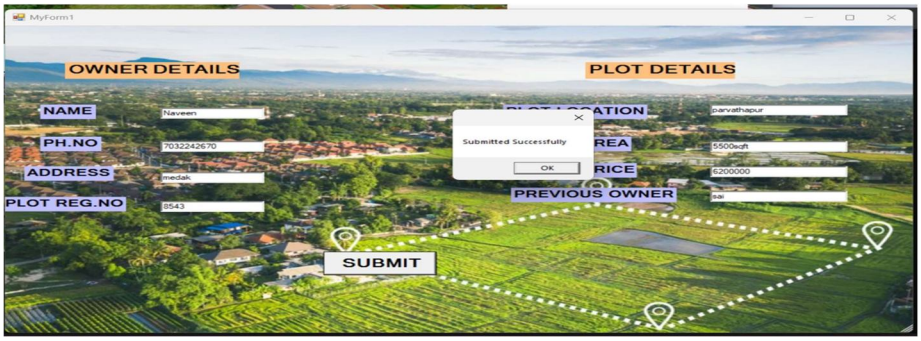


Fig. 4: Main Navigation Interface

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

B. Backend Implementation

The backend logic is embedded in event handlers within each form. SQL queries are dynamically constructed and executed to perform database operations.

1) Database Connectivity: Implemented using ADO.NET classes (SqlConnection, SqlCommand, SqlDataAdapter, SqlDataReader).

2) Key Backend Functions:

- Insert Operation (MyForm1): Executes INSERT queries for new plot registrations.
- Search Operation (MyForm2): Executes dynamic SELECT queries based on buyer filters.
- Update Operation (MyForm4): Executes UPDATE statements to modify existing data.
- Delete Operation (MyForm4): Executes DELETE queries using PlotRegNo.
- View Operation: Fetches and displays all records from the database.

3) Validation & Error Handling:

- Checks for empty fields and duplicates before insertion.
- Try-Catch blocks ensure stable error handling (e.g., connection errors).

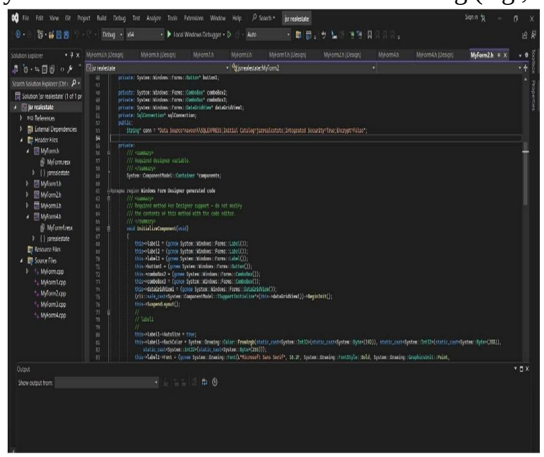


Fig. 5: Record Inserted Successfully

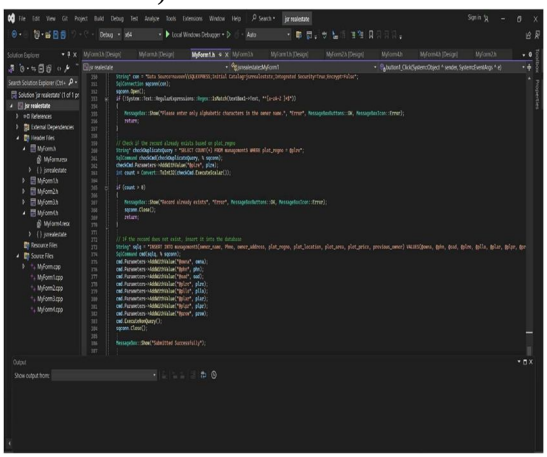


Fig. 6: Database Connection Successful

C. Database Implementation

The system database is implemented using Microsoft SQL Server.

1) Database: jsrrealestste

2) Main Table: management5

owner_name	Phno	owner_address	plot_regno	plot_location	plot_area	plot_price	previous_owner
			0			0	
naveen	7032242670	medak	4618	L.B Nagar	3500	5400000	praveen
Naveen	7032242670	medak	4656	Parvathapur	2500sqft	4400000	praveen
pavan	9032242656	medak	4674	Thamaka	2500sqft	3500000	deepak
pavan	9032242656	medak	4678	Banjarahills	5500sqft	8500000	srikanth
ram	9866509984	borgon	4732	banjarahills	4500sqft	5400000	sai
naveen	7032242670	warangal	4913	L.B Nagar	4000sqft	5400000	niraj
naveen	+917032242670	medak	5012	uppal	5000sqft	2500000	sandeep
hari	+919014132668	nizambad	5089	parvathapur	4500sqft	3500000	pradeep
shirisha	8496578452	uppal	5198	secunderabad	5500	4500000	nageshwar
arun	8473198565	uppal	5473	ECIL	5500	5800000	srikanth
mallesh	+919705339939	parvathapur	5696	ECIL	5500sqft	7500000	shekhar
naveen	7032242670	medak	5714	kukatpally	6000sqft	9000000	madhu
sandeep	9848336685	karimnagar	5791	parvathapur	4000sqft	5600000	naveen
sathwika	8469745896	nizampet	5814	banjarahills	5000sqft	5500000	shivaram

Fig. 7: SQL Server Table “management5” in jsrrealestste Database

D. Key Modules Implemented

1) User/Owner Module (MyForm1)

- Owners register plots for sale by entering details.
- Data is validated and stored in the database.

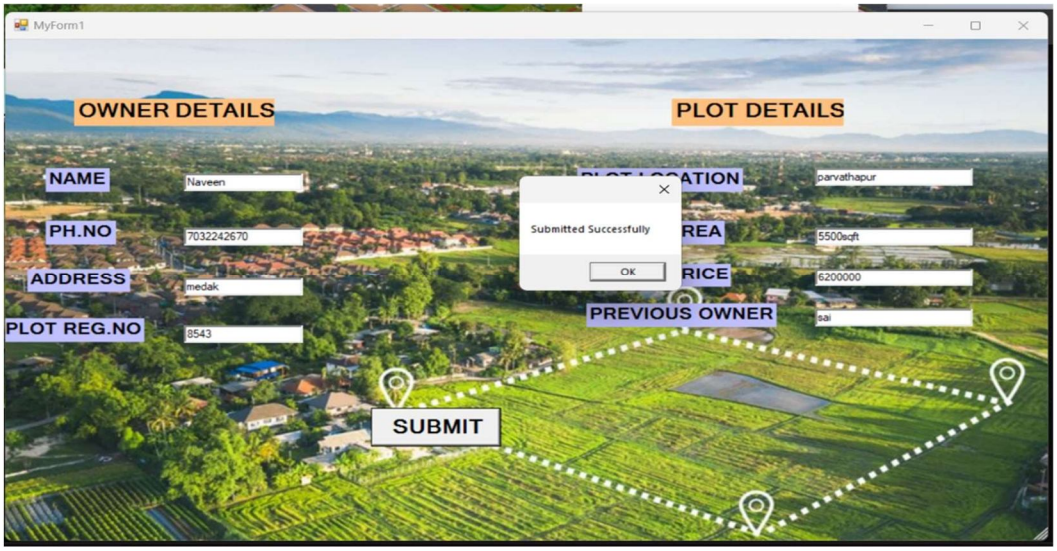


Fig. 8: Adding New Plot

2) Buyer/Search Module (MyForm2)

- Buyers search available plots by location and area.
- Results displayed in DataGridView.

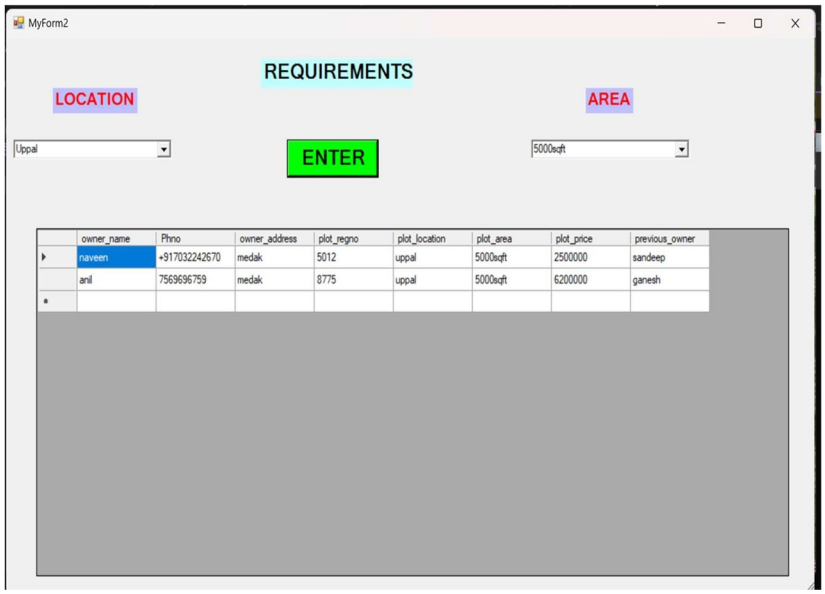


Fig. 9: Searching Plots by Location and Area

3) Management Module (MyForm3 & MyForm4)

- Update existing records (price, location, etc.).
- Delete records using PlotRegNo.
- Search records efficiently.

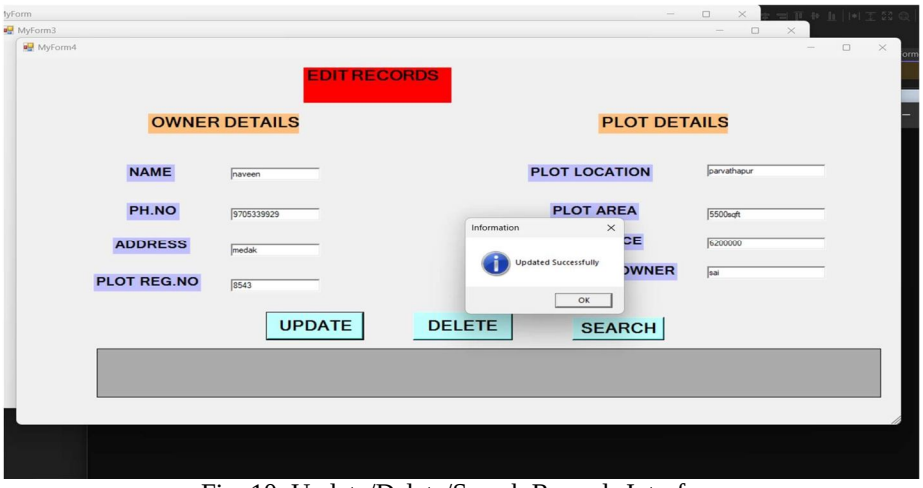
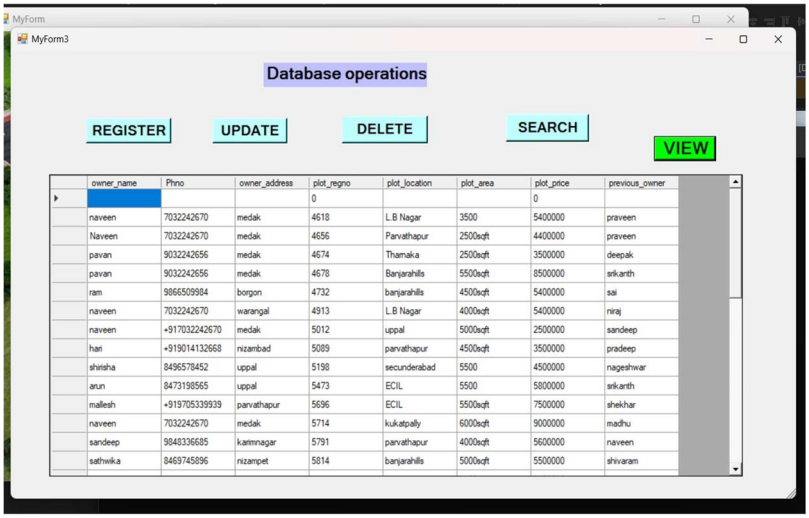


Fig. 10: Update/Delete/Search Records Interface

4) Data Display Module

- View all property listings from the database.
- Provides administrators a complete overview.



owner_name	Phno	owner_address	plot_regno	plot_location	plot_area	plot_price	previous_owner
			0			0	
naveen	7032242670	medak	4618	L.B Nagar	3500	5400000	praveen
Naveen	7032242670	medak	4656	Parvathapur	2500sqft	4400000	praveen
pavan	9032242656	medak	4674	Thamaka	2500sqft	3500000	deepak
pavan	9032242656	medak	4678	Banjarahills	5500sqft	8500000	srikanth
ram	9866509984	borgon	4732	banjarahills	4500sqft	5400000	sai
naveen	7032242670	warangal	4913	L.B Nagar	4000sqft	5400000	niraj
naveen	+917032242670	medak	5012	uppal	5000sqft	2500000	sandeep
hari	+919014132668	nizambad	5089	parvathapur	4500sqft	3500000	pradeep
shirisha	8496578452	uppal	5198	secunderabad	5500	4500000	nageshwar
arun	8473198565	uppal	5473	ECIL	5500	5800000	srikanth
mallesh	+919705339939	parvathapur	5696	ECIL	5500sqft	7500000	shekhar
naveen	7032242670	medak	5714	kukatpally	6000sqft	9000000	madhu
sandeep	9848336685	karimnagar	5791	parvathapur	4000sqft	5600000	naveen
sathwika	8469745896	nizampet	5814	banjarahills	5000sqft	5500000	shivaram

Fig. 11: Displaying All Records in DataGridView

E. Testing and Validation

Unit Testing

- Each button event (Insert, Search, Update, Delete) was tested with valid and invalid inputs.

Functional Testing

- Verified workflows such as registering a plot, buying/searching, updating, and deleting records.

Error Handling Testing

- Invalid entries (e.g., empty fields, wrong formats) were tested to ensure proper error messages.

VI. TECHNOLOGY AND STACK OVERVIEW

The Web-Based Real Estate Management System (WREMS) is developed using a combination of modern web technologies that ensure seamless property management, secure communication, and efficient database handling. The system leverages HTML, CSS, JavaScript, PHP, and MySQL to create a robust and user-friendly platform.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VIII Aug 2025- Available at www.ijraset.com

A. HTML (Hyper Text Markup Language)

HTML is the standard markup language used to structure and present content on the web.

- Advantages for WREMS:
  - Provides a clear and organized structure for property listings, forms, and dashboards.
  - Easily integrates with CSS and JavaScript for enhanced functionality and design.
  - Universal browser support ensures accessibility for all users.
- Usage in WREMS: Used to design the property listing pages, registration/login forms, and buyer-seller dashboards.

B. CSS (Cascading Style Sheets)

CSS is used to control the layout and styling of the application.

- Advantages for WREMS:
  - Enhances user experience through responsive and visually appealing design.
  - Provides consistent styling across different devices and screen sizes.
  - Facilitates separation of design and content for easier maintenance.
- Usage in WREMS: Implements responsive designs for property galleries, forms, and navigation menus, ensuring the platform is mobile-friendly.

C. JavaScript

JavaScript adds interactivity and dynamic features to the system.

- Advantages for WREMS:
  - Enables dynamic property filtering, search, and form validation.
  - Enhances responsiveness without requiring page reloads.
  - Provides seamless interaction between users and the system.
- Usage in WREMS: Handles client-side validation, dynamic property search filters, and real-time updates in buyer-seller communication.

D. PHP (Hypertext Preprocessor)

PHP is a widely used open-source server-side scripting language for backend development.

- Advantages for WREMS:
  - Simplifies database integration with MySQL.
  - Provides secure user authentication and session handling.
  - Efficient for handling form submissions and property management logic.
- Usage in WREMS: Manages user authentication, property posting, buyer-seller messaging, and backend business logic.

E. MySQL

MySQL is a relational database management system (RDBMS) that stores and manages system data.

- Advantages for WREMS:
  - Provides structured storage of property listings, user profiles, and communication records.
  - Ensures data integrity with relational constraints.
  - Supports efficient querying, indexing, and transaction management.
- Usage in WREMS: Stores property details, buyer and seller information, messages, and transaction history in relational tables.

F. Additional Tools and Libraries

- Bootstrap: For responsive and consistent UI components.
- AJAX: For asynchronous communication between frontend and backend.
- phpMyAdmin: For easy database administration and management.
- XAMPP/WAMP Server: Provides a local development environment for PHP and MySQL integration.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VIII Aug 2025- Available at www.ijraset.com

VII. RESULTS AND DISCUSSION

The Web-Based Real Estate Management System (WREMS) was evaluated based on functionality, usability, performance, and scalability. The system was tested using multiple user roles, including Admin, Property Owner, Buyer, and Agent, across different property listing, searching, and communication scenarios.

A. Functional Performance

The system successfully executed all core functions, including:

1) Secure login and registration with role-based access control.
2) Property listing with details such as location, price, size, and amenities.
3) Search and filter functionality enabling buyers to find properties based on criteria.
4) Buyer-seller communication through integrated messaging.
5) Database-backed storage ensuring consistency of listings and transactions.

During testing, the average response time for database queries and API calls was below 300 ms, ensuring a smooth browsing and communication experience.

B. Usability Evaluation

A usability survey was conducted among 20 participants, including property owners, potential buyers, and agents. The feedback highlighted:

1) Simple and intuitive navigation for property search and listing.
2) Minimal learning curve due to a clean and structured interface.
3) Cross-device compatibility (desktop, tablet, and mobile).

TABLE I
SURVEY RESULTS TABLE

Criteria	**Average Score (Out of 5)**
Ease of Use	4.6
Response Time	4.7
Visual Design	4.5
Overall Satisfaction	4.7

C. Comparison With Existing Systems

When compared with existing real estate portals [1][2], WREMS demonstrated notable advantages:

1) Direct Buyer-Seller Communication: Unlike many platforms that rely on third-party intermediaries, WREMS allows secure in-platform communication between buyers and sellers.
2) Customizable Search Filters: Advanced search parameters (price range, property type, location, and amenities) provide more refined results.
3) DBMS-Backed Storage: Ensures data consistency and efficient retrieval compared to spreadsheet-based or semi-structured solutions.

D. Scalability and Future Enhancements

The system's modular web-based architecture supports horizontal scaling, allowing independent management of frontend, backend, and database services.

Future enhancements include:

1) AI-Powered Property Recommendation System to suggest properties based on buyer preferences.
2) Integration with Payment Gateways for secure booking and advance payments.
3) Mobile Application for real-time property alerts and messaging.
4) Geo-Location Mapping for visual property exploration.

©IJRASET: All Rights are Reserved

VIII. CONCLUSION

The Web-Based Real Estate Management System (WREMS) presented in this paper demonstrates how modern web technologies and database management systems can effectively address the limitations of traditional real estate practices. By implementing the system with a structured DBMS-backed approach and an intuitive user interface, WREMS provides a comprehensive platform that simplifies property listings, enhances buyer–seller communication, and ensures secure data handling.

Compared to conventional property management methods, WREMS introduces centralized data storage, advanced search and filtering, and direct in-platform messaging, thereby significantly improving efficiency, accessibility, and transparency. The system's SQL Server database enables reliable and scalable management of large volumes of property data, while the Visual Studio development environment supports robust frontend design, debugging, and seamless integration with backend services.

The emphasis on user-centered design and communication fosters trust and confidence among stakeholders, empowering buyers and sellers to make informed decisions. Usability testing results confirm that the platform's clean design, responsive interface, and role-based access control minimize complexity while ensuring data security.

Looking ahead, the system can be extended with features such as AI-driven property recommendations, payment gateway integration, and mobile applications to further enhance user convenience and broaden accessibility. This project exemplifies how web-based applications, backed by DBMS technology, can modernize traditional industries, bridging the gap between digital transformation and practical real-world applications in real estate management.

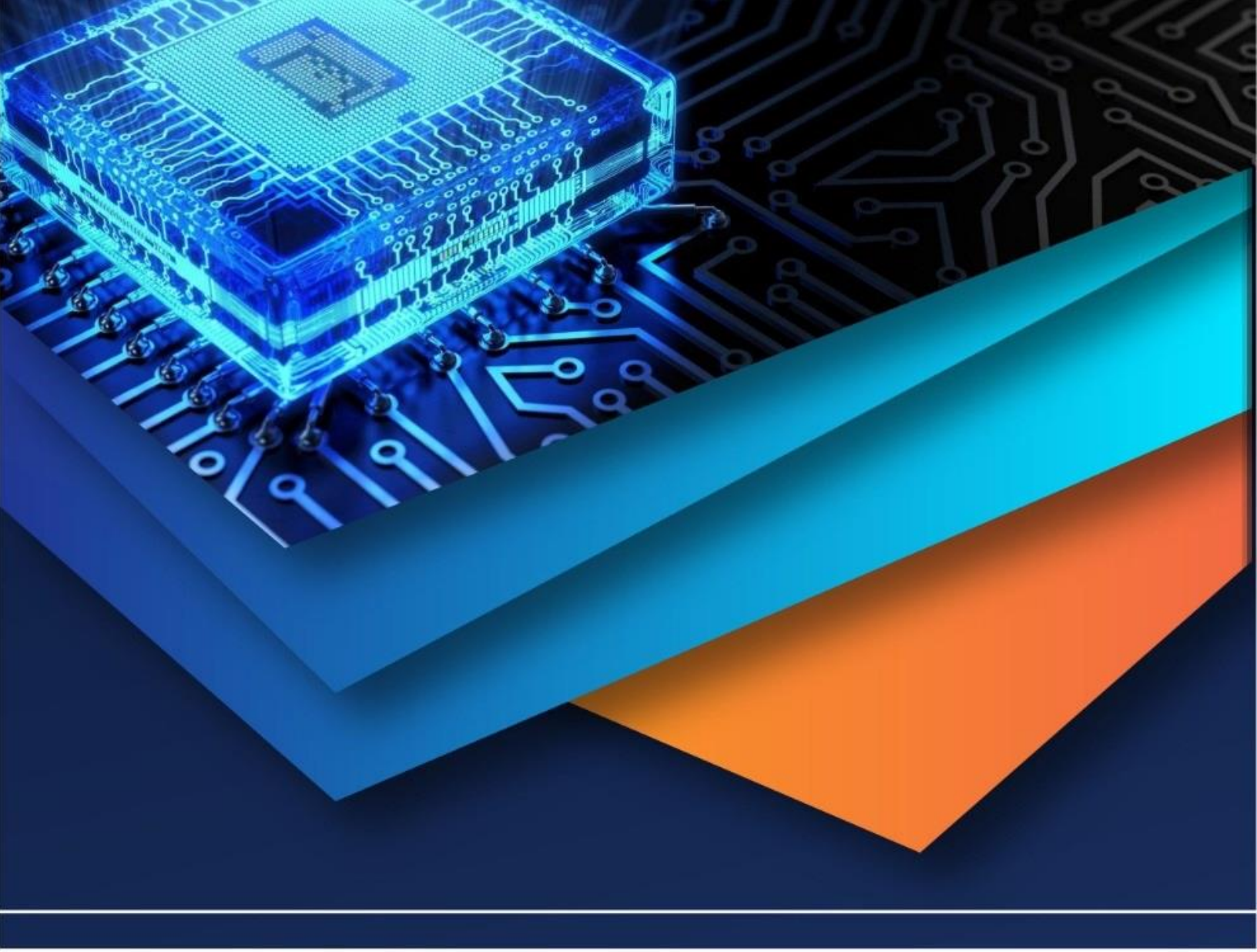
ACKNOWLEDGEMENT

I would like to express my sincere gratitude to Mr. Pranay Vemula, Assistant Professor, School of Engineering, Department of CSE, Aurora Deemed to be University, for his invaluable guidance, support, and encouragement throughout the development of this project. I also extend my thanks to my faculty members and peers who provided constructive feedback during the testing phase. Finally, I acknowledge Aurora Deemed to be University for providing the resources and platform to carry out this research and implementation successfully.

REFERENCES

[1] Singh, A., & Verma, R., "Web-Based Real Estate Listing Systems: A Survey," International Journal of Real Estate Management, vol. 8, no. 2, pp. 45–53, 2021.
[2] Kumar, S., & Reddy, M., "Efficiency of Buyer-Seller Platforms in Real Estate Using Database Systems," Journal of Property Technology, vol. 5, no. 4, pp. 98–105, 2022.
[3] Patel, T., & Das, N., "Design and Implementation of Real Estate Listing Portals with Messaging Features," Proceedings of the International Conference on Web Applications, pp. 120–128, 2023.
[4] Microsoft, "SQL Server Documentation," [Online]. Available: https://learn.microsoft.com/sql/
[5] Microsoft Visual Studio Team, "Visual Studio Documentation," [Online]. Available: https://learn.microsoft.com/visualstudio/
[6] Johnson, P., "Secure Database-Driven Web Applications: Best Practices," Computers & Security Journal, vol. 37, pp. 15–24, 2020.
[7] Tan, L., "Role-Based Access Control in Web Applications," Journal of Information Access, vol. 12, no. 1, pp. 30–38, 2019.
[8] Zhao, K., "Implementing CRUD Interfaces with Windows Forms and SQL Server," International Journal of Software Engineering, vol. 14, no. 3, pp. 67–75, 2022.
[9] Singh, V., "Evaluating Usability in Property Management Software: A Survey Study," UX in Practice Journal, vol. 3, no. 2, pp. 54–60, 2023.
[10] Garcia, R., "Responsive Design Techniques for Desktop and Mobile Forms," Proceedings of the UI/UX Design Conference, pp. 88–95, 2021.

©IJRASET: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)