

“Research Paper on Home Craft: A Virtual Architect”

1st Prof. Aditya Patil
DEpt.Of Computer Science
&Engineering Vidarbha Insitute of
Technology, Nagpur
Email

3rd Mr. PRATYUSH DUBEY
DEpt.Of Computer Science
&Engineering Vidarbha Insitute of
Technology, Nagpur

4th Ms. JANHVI WANKHEDE
DEpt.Of Computer Science
&Engineering
Vidarbha Insitute of Technology.Nagpur

2nd Mr. RONIT PATNE
DEpt.Of Computer Science
&Engineering
Vidarbha Insitute of Technology,
Nagpur

5th. MR. Ayan Khan
DEpt.Of Computer Science
&Engineering
Vidarbha Insitute of Technology.Nagpur

Abstract— The rapid advancement of web technologies has created opportunities to transform traditionally complex desktop-based applications into lightweight, accessible browser-based systems. Computer-Aided Design (CAD), which has long relied on high-performance software and skilled users, remains largely inaccessible for quick, intuitive 2D floor planning tasks. Existing solutions fall into two extremes: powerful desktop CAD tools that are complex, resource-intensive, and require technical expertise, and simple web-based planners that prioritize usability but lack geometric precision and intelligent automation. This gap highlights the need for a system that can balance accuracy, performance, and ease of use within a single platform.

This project presents a web-based 2D layout planning system that leverages vector-based rendering, event-driven architecture, and computational geometry to provide a real-time, interactive design environment. The system is built on an SVG canvas mapped to a Cartesian coordinate system, where all elements such as walls, rooms, and objects are represented mathematically. User interactions are processed through a mode-based state mechanism, allowing dynamic interpretation of inputs for different operations such as wall drawing, object placement, and room configuration. Advanced geometric algorithms are integrated into the core engine, including line intersection detection, nearest-point snapping, ray casting for spatial identification, and polygon-based room detection.

A key contribution of the proposed system is its ability to automatically detect closed wall structures and convert them into room entities using graph traversal techniques. The system further enhances usability by calculating room areas in real time using the Shoelace formula and providing continuous visual feedback during the design process. Unlike traditional CAD tools, which rely on manual validation and discrete operations, this system ensures continuous geometry processing, the use of a structured JSON-based data model allows efficient state management, enabling features such as undo/redo functionality and seamless data persistence through browser local storage.

1. INTRODUCTION

The evolution of digital design tools has significantly transformed the architectural and interior design industries. Traditionally, Computer-Aided Design (CAD) systems have been confined to high-performance desktop environments, requiring specialized hardware, licensed software, and trained professionals. While these tools provide precision and flexibility, they introduce barriers in terms of accessibility, cost, and usability.

With the advancement of web technologies such as HTML5, SVG, and modern JavaScript engines, it is now possible to shift computationally intensive graphical applications into browser environments. This transition is not merely a technological shift but represents a paradigm change in how design tools are accessed and utilized. The proposed system, Dream Layout Planner, is a browser-based CAD application that focuses on 2D architectural floor planning. It leverages vector-based rendering techniques and computational geometry to provide a real-time, interactive drafting experience. Unlike raster-based systems, which depend on pixel manipulation, this application models every element mathematically using coordinate geometry.

A key motivation behind this project is to eliminate inefficiencies in traditional workflows. In conventional CAD tools, tasks such as wall alignment, area measurement, and object placement often require manual intervention. This not only increases design time but also introduces the possibility of human error.

The proposed system automates these operations using algorithms such as line intersection detection, snapping heuristics, and polygon identification.

Furthermore, the application adopts an event-driven architecture, where user interactions (mouse movement, clicks, keyboard inputs) dynamically influence the system state. This approach ensures responsiveness and enables real-time feedback, which is critical in design environments.

Another important aspect is accessibility. By operating entirely within a browser, the system removes the need for installation and allows cross-platform usage. This makes it suitable not only for professionals but also for students and casual users who require quick and efficient layout visualization.

2. LITERATURE REVIEW

Literature indicates a clear need for a system that combines the precision of traditional CAD tools with the accessibility of web-based platforms, while incorporating real-time geometry processing and intelligent automation. Recent studies highlight the importance of vector-based rendering techniques, particularly SVG, in building scalable and precise web-based design applications. Unlike raster graphics, vector graphics represent elements mathematically, enabling accurate manipulation and infinite scalability without loss of quality. This makes them highly suitable for CAD-like applications where precision is critical.

□ Ensures resolution-independent rendering for consistent visualization at any zoom level Enables direct manipulation of individual elements through DOM integration

Literature on human-computer interaction emphasizes that real-time feedback is essential for improving usability and reducing user errors in interactive systems. In design applications, immediate visual and numerical feedback allows users to make informed decisions during the creation process rather than relying on post-validation.

Literature indicates a clear need for a system that combines the precision of traditional CAD tools with the accessibility of web-based platforms, while incorporating real-time geometry processing and intelligent automation. Recent studies highlight the importance of vector-based rendering techniques, particularly SVG, in building scalable and precise web-based design applications. Unlike raster graphics, vector graphics represent elements mathematically, enabling accurate manipulation and infinite scalability without loss of quality. This makes them highly suitable for CAD-like applications where precision is critical.

□ Ensures resolution-independent rendering for consistent visualization at any zoom level Enables direct manipulation of individual elements through DOM integration

Literature on human-computer interaction emphasizes that real-time feedback is essential for improving usability and reducing user errors in interactive systems. In design applications, immediate visual and numerical feedback allows users to make informed decisions during the creation process rather than relying on post-validation.

3. PROBLEM STATEMENT

The Traditional CAD systems used for 2D floor planning are powerful but impractical for a large segment of users due to their complexity, heavy resource requirements, and steep learning curve. These tools are designed with a focus on flexibility and control, assuming that the user possesses technical expertise to manually handle geometry, constraints, alignment, and validation. As a result, even simple tasks such as drawing walls, aligning components, or calculating room areas become time-consuming and error-prone. On the other hand, lightweight web-based floor planning tools improve accessibility but lack the computational intelligence required for accurate design, as they often rely on basic drawing mechanisms without real-time validation, intelligent snapping, or automatic room detection.

This creates a clear gap between high-end CAD systems that are too complex and basic tools that are too limited. Therefore, there is a need for a system that combines the precision of traditional CAD with the simplicity and accessibility of web-based platforms, while incorporating real-time geometry processing and intelligent automation to reduce user dependency and improve design accuracy. Most existing systems do not provide proactive assistance during the design process. Tasks such as alignment, snapping, validation, and area calculation are either manual or occur after the design is completed. The absence of real-time feedback and intelligent automation results in inconsistent layouts and requires repeated corrections, increasing both time and effort. This highlights the need for systems that can guide users dynamically and prevent errors during creation itself. There is a clear gap in the current ecosystem where high-end CAD tools are too complex and resource-intensive, while lightweight web-based planners lack precision and advanced features. Users are forced to choose between accuracy and simplicity, as no existing solution effectively combines both. This creates a demand for a balanced system that offers CAD-level precision with a user-friendly, accessible interface suitable for real-time usage.

4. Proposed System:

The proposed system replaces the rigid, step-by-step workflow of traditional CAD tools with an event-driven architecture based on dynamic modes. Instead of forcing the user to follow a fixed sequence, the system operates through context-aware modes such as wall drawing, object placement, and room configuration. User actions like mouse movement, clicks, and keyboard inputs are continuously captured and interpreted in real time based on the active mode. This approach eliminates unnecessary steps and allows users to interact naturally with the system, making the design process faster, more intuitive, and less dependent on technical expertise.

Unlike existing systems where geometry updates only after manual intervention, the proposed system continuously processes geometric relationships in real time. Every movement of the cursor triggers instant recalculation of distances, alignments, and intersections. When a wall is moved or modified, connected elements automatically adjust to maintain structural consistency. This creates a continuous feedback loop where errors are prevented during the design process itself rather than being detected later. As a result, the workflow becomes smooth, efficient, and significantly less error-prone.

The system introduces advanced snapping mechanisms based on mathematical calculations rather than simple grid alignment. Using techniques such as perpendicular distance calculation and nearest-point detection, the system automatically aligns walls and objects with high precision. Architectural elements like doors and windows are not just placed visually but are intelligently embedded into wall structures, respecting thickness and orientation. This removes the need for manual fine-tuning and ensures that all components are placed accurately, reducing inconsistencies in the layout.

The proposed system uses a structured JSON-based data model to represent all elements such as walls, rooms, and objects. This allows efficient storage, easy retrieval, and seamless integration with web technologies. Unlike proprietary formats, this approach ensures portability and flexibility. It also enables robust state management features such as undo and redo by storing snapshots of the design at different stages. This improves reliability and allows users to experiment freely without the risk of losing progress.

The system shifts from a tool-centric approach to a user-centric one by embedding intelligence into core operations. Tasks that were previously manual—such as alignment, validation, and measurement—are now handled automatically by the system. This reduces cognitive load on the user and minimizes the chances of human error. Instead of requiring expert-level knowledge, the system guides the user implicitly through automated corrections and suggestions, making it suitable for both professionals and beginners.



6. Conclusion

The developed 2D layout planner demonstrates that complex CAD functionality can be effectively implemented within a lightweight, browser-based environment without relying on heavy desktop software. By integrating vector-based rendering, event-driven interaction, and real-time geometry processing, the system ensures both precision and usability. Unlike traditional CAD tools that depend heavily on user expertise, this system shifts key responsibilities such as alignment, snapping, and validation to

automated algorithms, reducing human error and improving workflow efficiency.

The system successfully addresses major limitations of existing methodologies by introducing continuous feedback, intelligent snapping, and automatic room detection. Features such as real-time area calculation and structured data handling further enhance the reliability and practicality of the application. As a result, the tool is not only suitable for professionals but also accessible to beginners who require a simplified yet accurate platform for floor plan design. This balance between technical accuracy and ease of use is the core strength of the proposed system.

While the current system provides a strong foundation for 2D floor planning, it can be further enhanced to meet industry-level requirements. One major improvement would be the integration of 3D visualization using technologies like WebGL, allowing users to convert 2D layouts into interactive 3D models. Additionally, implementing cloud storage and multi-user collaboration features would enable real-time teamwork and accessibility across devices, making the system more practical for real-world usage. Further advancements can include support for standard file formats such as DWG or DXF to improve interoperability with existing CAD tools. The system can also be extended with AI-based features for automated layout suggestions, cost estimation, and optimization of space utilization. With these enhancements, the application has the potential to evolve from an academic project into a scalable, industry-ready solution for architectural design and planning.

REFERENCES

- [1] W3C, “Scalable Vector Graphics (SVG) 2 Specification,” 2018. [Online]. Available: <https://www.w3.org/TR/SVG2/>
- [2] Mozilla Developer Network (MDN), “SVG Tutorial,” 2024. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/SVG>
- [3] Mozilla Developer Network (MDN), “JavaScript Guide,” 2024. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide>
- [4] Bootstrap Team, “Bootstrap 5 Documentation,” 2024. [Online]. Available: <https://getbootstrap.com/docs/5.0/getting-started/introduction/>
- [5] J. D. Foley, A. van Dam, S. K. Feiner and J. F. Hughes, Computer Graphics: Principles and Practice, 3rd ed. Boston, MA, USA: Addison-Wesley, 2014.
- [6] M. de Berg, O. Cheong, M. van Kreveld and M. Overmars, Computational Geometry: Algorithms and Applications, 3rd ed. Berlin, Germany: Springer, 2008.
- [7] E. W. Weisstein, “Shoelace Formula,” MathWorld. [Online]. Available: <https://mathworld.wolfram.com/ShoelaceFormula.html>
- [8] GeeksforGeeks, “Line Intersection in Computer Graphics,” 2024. [Online]. Available: <https://www.geeksforgeeks.org/line-intersection-in-computer-graphics/>
- [9] GeeksforGeeks, “Point in Polygon using Ray Casting Algorithm,” 2024. [Online]. Available: <https://www.geeksforgeeks.org/point-in-polygon-in-cpp/>