iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** XI **Month of publication:** November 2026

DOI: https://doi.org/10.22214/ijraset.2026.85513

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 14 Issue XI Nov 2026- Available at www.ijraset.com

DreamHome AI: A Rule-Based and Constraint-Based Home Construction Plan Designer with Parking Planning and Cost Estimation

Kavithra R[1], Lohitha R[2], Mounika K[3]
Department of Computer Science and Engineering, Adhiyamaan College of Engineering (Autonomous), Hosur, Tamil Nadu, India

Abstract: ***Preliminary home planning is usually manual and fragmented: land details, room layout, parking and cost are handled separately by architects, draftsmen and contractors, so every change in requirements forces redrawing and recalculation. This paper presents DreamHome AI, a web-based home construction plan designer that brings requirement validation, 2D floor-plan generation, parking planning and construction cost estimation into one workflow. A constraint-based algorithm validates the user's inputs, a rule-based spatial allocation algorithm arranges the rooms within the plot, and geometric checks test room dimensions, plot boundaries and overlapping placements. Cost is estimated from the built-up area and a city-specific construction rate. The application is implemented with a Next.js and TypeScript frontend, a Python FastAPI backend and a PostgreSQL database. A worked example for a 30 ft × 40 ft east-facing plot produces a two-floor, three-bedroom layout and a cost estimate for three quality tiers. The generated plans and estimates are preliminary and must be reviewed by qualified professionals before construction.***

Keywords: ***home construction planning, floor-plan generation, rule-based spatial allocation, constraint-based validation, parking planning, construction cost estimation, FastAPI, Next.js.***

I. INTRODUCTION

Building a house begins with questions that are easy to ask and slow to answer. Will the required rooms fit on the plot? Is there space left for a car? What will it cost? In the present process a home owner approaches an architect, a draftsman or a contractor, who prepares paper sketches or drawings, and a separate estimate is prepared, often in a spreadsheet, from experience and approximate rates. Land details, room arrangement, parking and cost are kept in different places, and every change in requirements means redrawing the plan and recalculating the cost. The owner often learns that a plan is unaffordable only after much effort has been spent on it.

This paper presents DreamHome AI, a web application that automates the preliminary stage of this process. The user enters plot dimensions, facing direction, number of floors, rooms and parking needs. The system validates the inputs, generates a 2D floor plan, plans parking, and estimates the construction cost, and it stores the results as a project that can be reviewed and regenerated. The contributions are:

1) An integrated workflow that links land assessment, requirement capture, floor-plan generation, parking planning and cost estimation;
2) A planning engine that combines planning rules (relative placement of rooms) with feasibility constraints (fit, minimum size, no overlap);
3) A city-based, tiered construction cost estimate tied to the built-up area of the generated plan; and
4) A working implementation with a Next.js frontend, FastAPI backend and PostgreSQL storage.

The work is limited to preliminary residential planning. The planning engine is rule-based and constraint-based; it is not a trained machine-learning model, and the generated plans are not certified architectural drawings.

II. RELATED WORK

The literature reviewed gives the foundations for each part of the system but does not offer one complete application. Russell and Norvig describe intelligent agents, search and constraint-based problem solving [1], which underlie the idea of placing rooms subject to constraints. Deep learning and applied machine learning [2], [3] could later learn layout preferences from existing plans, but they need suitable datasets and do not by themselves guarantee that a plan satisfies geometric constraints.

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue XI Nov 2026- Available at www.ijraset.com

Computer vision [4] may help analyse existing plan images in future work. Building Information Modeling [5] and building construction principles [6] provide background on how building data and spatial organisation are structured, but neither provides a rule-based plan generation workflow for early planning. On the software side, FastAPI [7], Next.js [8], PostgreSQL [9] and SQLAlchemy [10] supply the web framework, interface, storage and database access, yet none of them contains planning algorithms.

The gap addressed here is therefore integration: no single reviewed resource combines requirement processing, automatic floor-plan generation, geometric validation, parking allocation and preliminary cost estimation in one application. DreamHome AI combines these functions in one web-based platform.

III. EXISTING AND PROPOSED SYSTEMS

A. Existing Process

The existing workflow is a loose sequence of steps: collecting land details, explaining requirements to a professional, sketching, reviewing, redrawing after each change, fitting parking into whatever space remains, and estimating cost at the end. If the estimate exceeds the budget, requirements are reduced and the cycle repeats. General-purpose tools (paper, drafting software, spreadsheets, messages) each hold only part of the information, so values must be carried by hand from one tool to the next; for example, if a room is enlarged in the drawing, the built-up area in the spreadsheet must be corrected manually or the cost becomes wrong.

B. Proposed System

DreamHome AI places the same stages in one platform. Table I compares the two approaches.

TABLE I
COMPARISON OF EXISTING AND PROPOSED SYSTEMS

Aspect	Existing process	DreamHome AI
Planning stages	Separate people and tools	Land, requirements, layout, parking and cost in one platform
Layout	Manual sketching or drafting	Rule-based and constraint-based generation
Cost estimate	Rough, prepared separately	Computed from built-up area and city rate
Parking	Fitted in after the layout	Planned with the layout
Revision	Redraw and recalculate	Change inputs and regenerate
Records	Paper and scattered files	Projects stored in a database

The system is intended for preliminary planning. Structural design, electrical and plumbing drawings, drawings for statutory approval, detailed material estimation and 3D modelling are outside its scope.

IV. SYSTEM ARCHITECTURE

The system follows a three-layer client-server architecture (Fig. 1). The presentation layer, built with Next.js, TypeScript and Tailwind CSS, renders the registration and login pages, land and house requirement forms, the 2D floor-plan viewer, the cost interface, the project summary and the dashboards. The application layer is a Python FastAPI service that handles authentication and access control, project management, validation of land and house requirements, floor-plan generation requests, cost estimation and administrative data retrieval. The data layer is a PostgreSQL database that stores user accounts, projects, land and location details, room requirements, generated floor-plan data and cost estimates. The browser never accesses the database directly; every request passes through the application layer, which enforces validation, authentication and ownership checks so that a user can open only the projects associated with their account.

©IJRASET: All Rights are Reserved

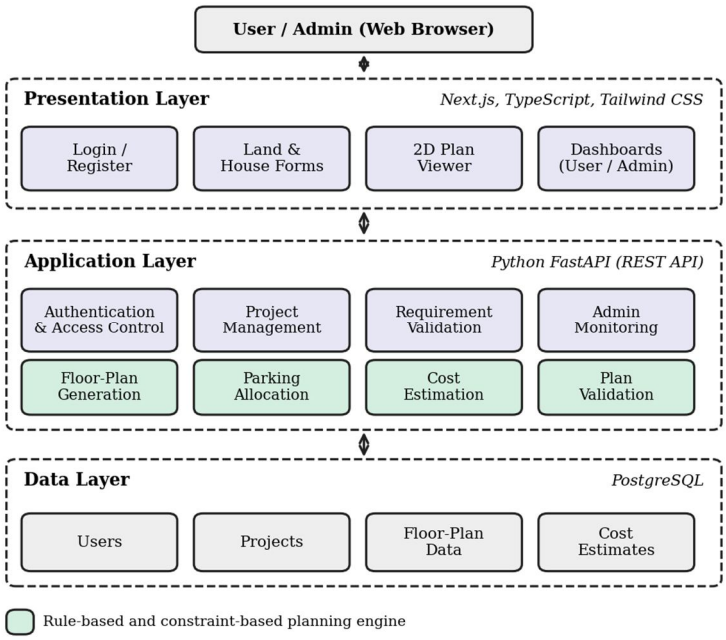


Fig. 1 Three-layer architecture of DreamHome AI: Next.js presentation layer, FastAPI application layer containing the planning engine, and PostgreSQL data layer

The main functional modules are user authentication, land assessment, house requirements, floor-plan generation, parking allocation, cost estimation, plan visualization, project management and administration. The planning engine is formed by the floor-plan generation, parking allocation and cost estimation modules together with plan validation.

V. METHODOLOGY

A. Requirement Validation

Before planning starts, each input is checked and invalid data is returned to the user for correction, so the planning engine is never run on meaningless input. The checks include: positive plot length and width; equal sides when a square plot is selected; at least one floor and at least one room; every room assigned to an existing floor; unique room names; total room area not exceeding the plot area; and parking area not exceeding the plot area. The same checks are applied again on the server because validation in the browser alone cannot be trusted.

B. Rule-Based Spatial Allocation

Rules express standard planning practice and decide the relative position and grouping of rooms. The rules used are:

- the living area is placed near the entrance so that visitors do not pass through private areas;
- bedrooms are grouped together, and bathrooms are placed close to the bedrooms they serve;
- the kitchen is placed near the dining area;
- the entrance and parking are placed on the road-access side of the plot; and
- the facing direction is considered when placing rooms, for daylight and ventilation.

C. Constraint Checking

Constraints decide whether a layout is feasible, and a layout that breaks one is not accepted. Every room must lie inside the buildable area; no two rooms on the same floor may overlap; each room must meet the minimum area and width for its type; the total room area on a floor must not exceed the buildable area of that floor; every room must be reachable from the entrance or a circulation space; and the parking area must remain on the accessible side of the plot. If a placement violates a constraint, the engine adjusts the placement or room proportions and tries again. If no feasible layout exists, the system reports which requirement could not be satisfied instead of producing an invalid plan.

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

D. Plan Representation and Validation

The generated plan is held as a structured object

$$P = \langle B, F, R, A \rangle \quad (1)$$

where B is the house boundary, F the floor-wise layout, R the room arrangement with dimensions, and A the designated areas such as parking and access. Before the plan is stored, the system checks that the boundary, floor-wise layout data, room arrangements and dimensions are present, and that the plan is associated with its project.

E. Parking Planning

Parking is planned together with the layout. The module takes the number of vehicles, calculates the corresponding space requirement, and checks that the space is accessible from the road side. If the free area is not enough, the module reports how many spaces can be provided so the user can reduce the built-up area or accept fewer spaces.

F. Cost Estimation

The cost is calculated from the built-up area of the generated plan:

$$C = A \times r + E \quad (2)$$

where C is the estimated cost, A the built-up area in square feet, r the construction rate per square foot for the selected city and quality tier, and E any additional cost ($E \geq 0$). For example, a built-up area of 1,800 sq. ft. at ₹2,000 per sq. ft. gives ₹36,00,000. The rate and the additional cost are configurable, so the estimate can follow market changes without changing the program. Because the estimate is computed from the same plan that is displayed, the plan and the cost cannot disagree, and the effect of adding a room or a floor on the budget is visible at once.

VI. IMPLEMENTATION

The frontend, written in Next.js and TypeScript with Tailwind CSS, submits the user's inputs to the FastAPI backend. Request and response data are described with Pydantic models, which enforce basic constraints such as positive dimensions. The backend applies the validation, planning and cost logic, and the results are stored in PostgreSQL and returned to the frontend for display. The main endpoints of the service are listed in Table II.

TABLE II
MAIN REST ENDPOINTS OF THE BACKEND

Endpoint	Function
POST /api/projects	Create a validated project
GET /api/projects/{id}	Retrieve a project
POST /api/projects/{id}/generate-plan	Validate requirements and generate the floor plan
GET /api/projects/{id}/floor-plan	Retrieve the stored plan
POST /api/projects/{id}/estimate-cost	Compute the cost estimate
GET /api/projects/{id}/summary	Project, plan and estimate together

Errors are returned as explicit responses: a missing project gives a not-found response, and a failed validation gives a bad-request response carrying the reason. For deployment, the server needs a 2.0 GHz dual-core processor, 4 GB RAM and 20 GB of storage (SSD recommended), and the client needs a modern browser on a device with at least 2 GB RAM and a stable internet connection.

VII. RESULTS AND DISCUSSION

The implemented system supports the complete workflow: registration and login, dashboard, land details, house requirements, floor and parking requirements, floor-plan generation, 2D visualization, cost estimation, project summary and an administrator dashboard. As an example, the land form is pre-filled with a standard 30 ft × 40 ft plot (1,200 sq. ft.), an east-facing road and a road width of 20 ft, 30 ft or 40 ft and above.

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

For an east-facing plot in Bengaluru with a ground-plus-one configuration, the system produced the layout in Fig. 2 for a three-bedroom duplex. The ground floor (880 sq. ft.) contains a car porch (10 × 15 ft), living hall (17 × 15.5 ft), parents' bedroom (12 × 14 ft), kitchen (10.5 × 12 ft), pooja room and bathroom. The first floor (880 sq. ft.) contains a master suite (14.5 × 16 ft), a children's bedroom (12 × 13.5 ft), a balcony deck (14.5 × 6 ft) and a work-from-home study (12 × 12 ft). The interface reports a floor area ratio of 1.46 against a limit of 1.75. The parking screen checks that a compact SUV and two two-wheelers fit in the designated space.



Fig. 2 Generated ground-floor and first-floor plans for the 30 ft × 40 ft east-facing example plot, with room names and dimensions in feet

The cost screen (Fig. 3) gives the estimate for three quality tiers at the Bengaluru rate index for Q4 2026, as summarised in Table III. For the expected tier, the RCC structure and masonry account for ₹22.8 lakh (48%), flooring, tiling and kitchen granite ₹8.1 lakh (17%), plumbing, sanitaryware and electrical work ₹7.1 lakh (15%), and doors, windows, painting and contingency ₹9.5 lakh (20%).

TABLE III
COST ESTIMATE BY QUALITY TIER

Tier	**Rate (₹/sq. ft.)**	**Estimate (₹ lakh)**
Conservative / Basic	1,850	41.2
Expected / Standard	2,150	47.5
Premium / Luxury	2,600	56.8

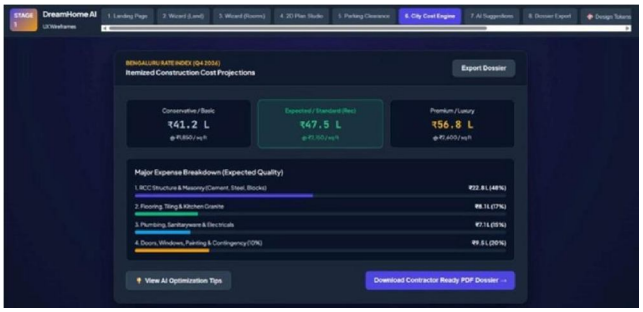


Fig. 3 City-based itemised construction cost projection for three quality tiers with expense breakdown for the expected tier

The results show the main benefit of integration: the owner enters the plot details and requirements once, receives the plan, the parking check and the cost together, and can change an input and regenerate. Since the cost is derived from the same plan that is shown, a change in the layout is reflected in the budget immediately.

VIII. LIMITATIONS

The system is a preliminary planning aid. It does not establish that generated plans are certified architectural drawings or that they comply automatically with every local building regulation, and structural design, soil analysis, foundation design, electrical and plumbing design, fire-safety compliance and statutory approvals remain outside its scope. The planning engine uses fixed rules and constraints and is not a trained machine-learning model. The accuracy of the cost estimate depends on how current the configured rates are, and actual cost may differ with material prices, labour charges, design changes and site conditions. A formal usability study and a comparison with plans drawn by professionals have not yet been carried out. Plans and estimates must be reviewed by qualified architects or engineers before construction, with reference to applicable codes such as the National Building Code of India [11].

IX. CONCLUSION AND FUTURE SCOPE

DreamHome AI integrates land measurement, house requirements, room configuration, floor-plan generation, 2D visualization, parking planning and cost estimation in a single web application. Rule-based and constraint-based logic turns user requirements into a structured plan, and an area-based, city-specific estimate gives an early view of the budget, reducing manual effort in preliminary planning. Future work includes machine-learning models to improve floor-plan generation, cost estimation using regional material prices and labour costs, multilingual support for Indian regional languages, better optimisation of room and parking layout, support for more plot sizes and residential requirements, and stronger project management and report generation.

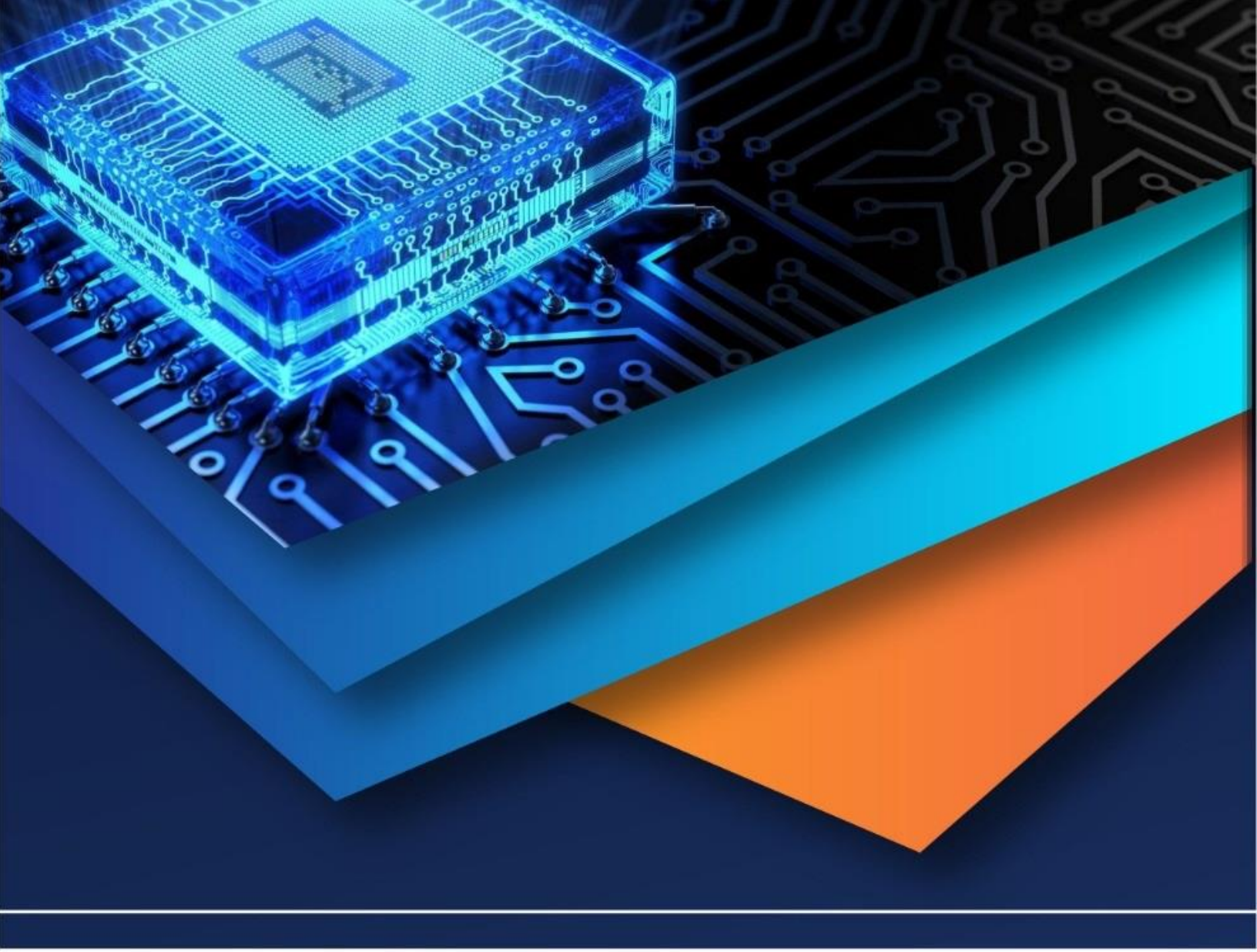
X. ACKNOWLEDGMENT

The authors thank the Principal, the Head of the Department of Computer Science and Engineering, the project supervisor, the project coordinator and the faculty of Adhiyamaan College of Engineering, Hosur, for their guidance and support.

REFERENCES

[1] S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 4th ed. Hoboken, NJ, USA: Pearson, 2020.
[2] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA, USA: MIT Press, 2016.
[3] A. Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 3rd ed. Sebastopol, CA, USA: O'Reilly Media, 2022.
[4] R. Szeliski, Computer Vision: Algorithms and Applications, 2nd ed. Cham, Switzerland: Springer, 2022.
[5] C. Eastman, P. Teicholz, R. Sacks, and K. Liston, BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors, 2nd ed. Hoboken, NJ, USA: Wiley, 2011.
[6] F. D. K. Ching, Building Construction Illustrated, 5th ed. Hoboken, NJ, USA: Wiley, 2014.
[7] FastAPI documentation. [Online]. Available: https://fastapi.tiangolo.com/
[8] Next.js documentation. [Online]. Available: https://nextjs.org/docs
[9] PostgreSQL Global Development Group, "PostgreSQL documentation." [Online]. Available: https://www.postgresql.org/docs/
[10] SQLAlchemy 2.0 documentation. [Online]. Available: https://docs.sqlalchemy.org/en/20/
[11] Bureau of Indian Standards, National Building Code of India 2016. New Delhi, India: BIS, 2016.

©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)