ISSN: 3107-6513

Smart Energy Saving System Using Presence and Ambient Light Based Intelligent Automation

A Software Simulation Approach for Intelligent Appliance Control

Dharani M[1], Aathira S N[2], Ashwathi S[3], Harish N[4], Pradeepa K[5]

[#]*Student,I Year Department of Computer Science with Artificial Intelligence and Data Science, KPR college of arts, science and research, Coimbatore, Tamilnadu, India.*

dharanimohanraj21@gmail.com , aathirasiva2324@gmail.com, ashsasi2009@gmail.com, harishn2008r@gmail.com

Abstract

This paper presents a software-based Smart Energy Saving System designed to demonstrate intelligent room-level energy control through simulated presence detection and ambient-light sensing. The system models the behaviour of a building in which artificial lighting should operate only when a space is occupied and the ambient light level is insufficient. The prototype is intentionally implemented as a front-end simulation, allowing the control logic, energy calculations, dashboards, logs, analytics, and user interaction to be evaluated without physical ESP32, passive infrared, or light-dependent resistor hardware. Four representative spaces are modelled: Living Room, Classroom, Office, and Bedroom. Each room maintains its own presence state, ambient-light value, appliance state, runtime, and energy consumption. The implemented light-control rule combines presence with a three-level ambient-light classification and automatically turns the light on only for an occupied dark room. Appliance energy is calculated from predefined power ratings and simulated runtime, while cost is estimated from a configurable electricity tariff. The user interface provides room monitoring, appliance status, energy analytics, activity logs, smart insights, and configuration controls. An accelerated simulation mechanism maps one second of real time to one simulated minute of appliance runtime, making the system suitable for demonstrations and controlled scenario analysis. This paper expands the prototype into a publication-oriented technical study, including architecture, algorithms, mathematical formulation, scenario-based evaluation, analytical energy comparisons, testing, limitations, and a pathway toward hardware deployment. The results in this paper distinguish clearly between project implementation behaviour, analytical calculations, and future real-world measurements.

Keywords: Smart energy management, occupancy-aware control, ambient-light sensing, PIR, LDR, intelligent automation, energy analytics, software simulation, appliance control, building energy efficiency

I. INTRODUCTION

Energy efficiency has become an important engineering objective because buildings and appliances consume energy continuously across residential, educational, commercial, and institutional environments. A small amount of unnecessary runtime repeated across many rooms can become a measurable operational cost. Lighting provides a particularly suitable target for automation because its useful operating state depends strongly on two contextual variables: whether a space is being used and whether the space already receives sufficient illumination.

Manual switching is simple but it transfers the control responsibility to occupants. In a classroom, office, corridor, laboratory, or bedroom, lights may remain active after occupants leave, while rooms with windows may remain artificially illuminated even when daylight is adequate. An automated controller can remove some of this dependence by converting environmental observations into repeatable rules. Research on occupancy-based lighting has shown that occupancy information can be useful for reducing lighting demand, while other work has explored occupant-centred approaches that balance energy reduction with user comfort [1], [2].

The proposed Smart Energy Saving System addresses this design problem through a software simulation. Instead of claiming that physical sensors have already been deployed, the project models the signals that would normally be obtained from a passive infrared sensor and a light-dependent resistor. The front-end implementation provides an interactive environment where a user can change room presence and ambient-light values and immediately observe the resulting control decision.

The project is broader than a single light-switch rule. It records appliance runtime, converts runtime into energy, estimates monetary cost, compares controlled operation against a continuous baseline, maintains an activity history, and produces rule-based smart insights. These functions make the prototype useful as an educational and pre-deployment engineering artefact: the control concept can be inspected before hardware is introduced, and the same data model can later be connected to real sensors and actuators.

In the remainder of the paper, the system is treated as a simulation-oriented smart building component. The paper first reviews related occupancy and lighting-control research, then explains the problem definition, architecture, control logic, user interface, mathematical model,

test methodology, analytical results, limitations, and a future hardware migration path. The central contribution is the transparent integration of sensing logic, energy calculation, monitoring, and decision support in a single software prototype.

II. RELATED WORK

Occupancy-based lighting control has been st,udied extensively because occupancy is a direct indicator of whether electric lighting may be required. De Bakker et al. reviewed occupancy-based lighting control in open-plan office spaces and identified occupancy patterns, control design, evaluation methods, and user factors as major elements influencing performance [1]. Their review highlights that potential savings are strongly affected by how the control strategy is designed and how occupants interact with it. This observation is relevant to the present project because a simulation should make the decision logic visible rather than treating the controller as a black box.

ISSN: 3107-6513

Nagy et al. proposed an occupant-centred lighting strategy that used occupancy information together with adaptive thresholds. Their experimental study demonstrated that lighting control can reduce energy use while considering occupant comfort [2]. The present prototype is simpler: it does not learn thresholds or adapt to individual occupants. Instead, it uses fixed ambient-light categories. This difference is intentional because the prototype is designed as an explainable starting point rather than a data-driven controller.

Research has also examined wireless sensors and actuators for occupancy-driven lighting in existing buildings. Labeodan et al. reported the use of low-cost wireless sensing and actuation to obtain occupancy information and control lighting [3]. Their work supports the future extension of this project into a hardware-and-network implementation.

Reviews of lighting-control technologies show that occupancy, time, daylight availability, and control strategy are common inputs in automated lighting systems [4]. More recent studies have examined building energy models and occupant-based controls in academic buildings, reinforcing the importance of space use and occupancy patterns when estimating savings [5]. Current smart-building research also considers IoT architectures, cyber-physical systems, analytics, and networked control as parts of next-generation building management [6], [7].

The distinction between the published literature and the present work is important. The literature contains field studies, simulation studies, adaptive control strategies, wireless sensing, and network architectures. The current project is a compact browser-based demonstration. It does not attempt to reproduce the full physical complexity of a building. Instead, it focuses on a transparent control rule and an observable dashboard. The paper therefore positions the prototype as a software simulation layer that can support teaching, hackathon demonstration, algorithm prototyping, and early-stage requirements analysis before hardware integration.

Study theme	Typical inputs	Primary focus	Implication for this project
Occupancy-based lighting [1]	Occupancy	Energy and control behaviour	Use presence as a control input
Occupant-centred control [2]	Occupancy + user response	Comfort and savings	Keep control logic understandable
Wireless sensing [3]	Networked sensors	Retrofit control	Plan hardware migration
Lighting technology review [4]	Occupancy + daylight + schedules	Technology comparison	Combine multiple context signals
Academic building control [5]	Occupancy + building model	Scenario-based energy analysis	Use analytical scenarios
Smart building / IoT [6], [7]	Sensors + controllers + analytics	System architecture	Separate sensing, decision and analytics layers

TABLE I. RESEARCH THEMES THAT MOTIVATE THE PROTOTYPE

III. PROBLEM STATEMENT, OBJECTIVES AND SCOPE

The engineering problem considered in this study is unnecessary electrical consumption caused by a mismatch between appliance state and actual room conditions. A light may be ON when no person is present, or a light may be ON even though the room is sufficiently bright. These situations represent avoidable runtime. A practical control system therefore needs at least one signal about occupancy and one signal about available illumination.

The project objective is to design a lightweight software simulation that demonstrates this contextual decision-making process and quantifies its effect using simple, transparent energy calculations. The system must be easy to interact with, must provide immediate visual feedback, and must retain enough state to support analytics over a short simulation session.

The main objectives are: (1) model multiple independent rooms; (2) represent presence and ambient-light inputs; (3) classify ambient light into dark, normal, and bright ranges; (4) automatically control lighting under automatic mode; (5) track appliance runtime and energy; (6) estimate monetary cost; (7) compare controlled operation against a continuous baseline; (8) maintain an activity log; (9) generate contextual insights; and (10) provide settings that can be changed without rewriting the application.

The scope is deliberately bounded. The implementation is a pure front-end simulation and therefore does not measure real voltage, current, illuminance, motion, or power factor. It does not directly control mains electricity. The fan is modelled as a manually operated appliance while the light is the primary automatically controlled load. The prototype also does not use machine learning, cloud services, or a physical database.

Within those boundaries, the project is suitable for early functional validation. It can answer questions such as whether the decision rule behaves consistently, whether energy calculations are internally coherent, whether multiple rooms can be managed independently, and whether the user interface exposes sufficient information for an operator. These are meaningful pre-hardware questions because errors in control logic are cheaper to correct in software than after wiring and deployment.

Objective	Implementation evidence	Current status
Multi-room modelling	Four independent room objects	Implemented
Occupancy input	Interactive present/absent state	Implemented
Ambient-light sensing	0–100 input with 3 classes	Implemented
Automatic light control	Presence + ambient-light rule	Implemented
Energy calculation	Runtime × power / 1000	Implemented
Cost estimation	Energy × tariff	Implemented
Monitoring	Room cards and virtual room	Implemented
Analytics	Consumed and saved energy chart	Implemented
Audit trail	Timestamped activity log	Implemented
Smart insights	Rule-based messages	Implemented
Physical hardware control	ESP32/PIR/LDR/relay	Future scope

TABLE II. OBJECTIVES AND CURRENT IMPLEMENTATION STATUS

IV. PROPOSED SYSTEM ARCHITECTURE

The architecture is organised as a set of logical modules rather than physical devices. The separation is important because the same control logic can later receive inputs from real hardware without requiring the user-interface layer to be redesigned. The architecture contains input models, a decision engine, appliance state, an energy engine, and an analytics and presentation layer.

Fig. 1. Proposed software architecture

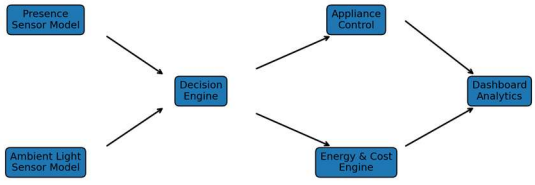


Fig. 1. Proposed software architecture.

The presence and ambient-light modules act as the sensing boundary. Presence is represented by a Boolean value, while ambient light is represented by a numerical percentage. The project classifies light values as dark for 0–30 percent, normal for 31–70 percent, and bright for 71–100 percent. These thresholds are simple enough to explain to a user and easy to modify during future experiments.

ISSN: 3107-6513

The decision engine applies the automation rule. In automatic mode, the light turns ON only when a person is present and ambient light is classified as dark. If a person is present but the light is normal or bright, the light remains OFF. If no person is present, the light is turned OFF regardless of ambient light. The rule therefore combines two conditions instead of relying on a single sensor.

The appliance layer stores light and fan states as well as runtime and energy values. The energy engine converts operating time into kilowatt-hours and calculates a corresponding monetary estimate from the configured tariff. The analytics layer then exposes totals, comparisons, trend charts, activity logs, and rule-based insights.

The architecture also includes persistent local state. Browser storage allows the simulation to retain settings and accumulated values between sessions. This is a useful software engineering choice for a prototype because it avoids introducing a server dependency while still demonstrating stateful behaviour.

V. SENSOR MODEL AND AUTOMATION LOGIC

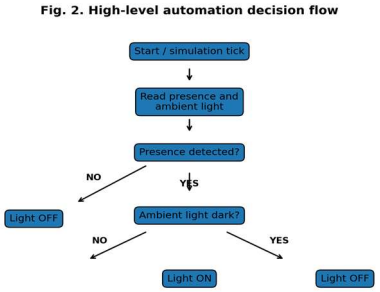


Fig. 2. High-level automation decision flow.

The current project emulates two common sensing mechanisms. The first is a passive infrared concept representing human presence. In the application, the user toggles the room between Present and Absent. The second is a light-dependent resistor concept represented by a slider from 0 to 100 percent. The slider provides continuous input while the decision layer converts it into one of three categories.

The separation between raw value and category is useful because a control rule rarely needs to react to every individual percentage point. It needs a meaningful state such as Dark, Normal, or Bright. The implemented classification function therefore maps the continuous value into three discrete states. This makes the rule easier to inspect and provides a clear explanation to users.

The core control rule can be expressed as follows. When presence is TRUE and ambient light is Dark, set the light state to ON. When presence is TRUE and ambient light is Normal or Bright, set the light state to OFF. When presence is FALSE, set the light state to OFF. If Automatic Mode is disabled, the system stops applying this rule and reports that manual control is active.

The current prototype intentionally uses a deterministic rule rather than a learning algorithm. This makes cause and effect easy to demonstrate: changing one input changes the decision immediately. The approach is suitable for a first-stage system because it provides a transparent baseline against which more advanced approaches could later be compared.

The fan has a different control policy in the current prototype. It is manually toggleable and contributes to the energy calculation when active. This distinction provides a useful demonstration that a smart energy system can manage heterogeneous appliances, with some devices automatic and others manual. Future versions can introduce temperature, comfort, or occupancy rules for fan control.

Presence	Ambient light	Automatic mode	Light action	Rationale
Present	Dark	ON	ON	Occupant needs artificial lighting
Present	Normal	ON	OFF	Ambient light may be adequate
Present	Bright	ON	OFF	Natural light is sufficient
Absent	Any	ON	OFF	No occupant is present
Any	Any	OFF	Manual	Automatic rule is disabled

TABLE III. DECISION RULE TRUTH TABLE

VI. SOFTWARE IMPLEMENTATION

Module	Primary responsibility	Key outputs
State manager	Maintain current and persistent state	Room objects, settings, chart arrays
Automation engine	Apply PIR + LDR decision rule	Light state, decision message
Simulation tick	Advance simulated time	Runtime, energy, charts
Energy engine	Calculate totals and savings	kWh, baseline, saved kWh, cost
Renderer	Synchronise UI with state	Cards, virtual room, metrics
Activity logger	Store system events	Timestamped event list
Insights engine	Generate contextual messages	Rule-based recommendations
Persistence layer	Retain state locally	Browser local storage

TABLE IV. SOFTWARE MODULES AND RESPONSIBILITIES

The one-tick simulation mechanism is a key design feature. One real-time second is treated as one simulated minute of appliance runtime. This acceleration allows a user to obtain readable energy values during a short demonstration while preserving the mathematical relationship between power, time, and energy. The dashboard therefore communicates that the numbers are simulated rather than physically metered.

Algorithm 1. Simulation tick sequence

Step	Operation	Purpose
1	Load current state	Preserve room and global values
2	Classify ambient light	Map 0–100% to Dark / Normal / Bright
3	Apply automation	Set light state from presence and light class
4	Accumulate runtime	Add one simulated minute when an appliance is ON
5	Update energy and cost	Convert runtime to kWh and monetary value
6	Refresh analytics and log	Maintain chart, insights and audit trail

TABLE XVII. SIMULATION ALGORITHM SUMMARY

VII. USER INTERFACE AND DASHBOARD DESIGN

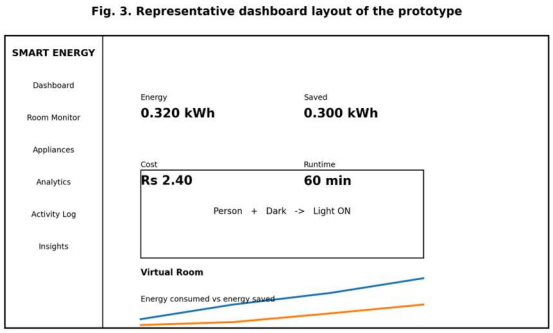


Fig. 3. Representative dashboard layout of the prototype.

The user interface was designed around the idea that an energy-management system should make both the current state and the reason for that state visible. The landing page introduces the system and emphasises presence, ambient light, automation, and savings. A Launch Dashboard action moves the user into the interactive simulation.

The main dashboard contains a presence panel, ambient-light panel, automation decision panel, summary statistics, a virtual room view, and a before-versus-after comparison. This arrangement creates a clear causal sequence: the user can see the input, read the decision, observe the virtual room, and then inspect the energy consequence.

The Room Monitor view provides a compact overview of the four simulated rooms. The user can switch between Living Room, Classroom, Office, and Bedroom. Each room has independent input and appliance state. The design therefore represents a small building rather than a single isolated room.

The Appliances view separates the light and fan. The light card reports its power rating, runtime, and energy used. The fan card provides similar information and includes a manual

ISSN: 3107-6513

ON/OFF control. A live power-draw section reports the sum of currently active appliance power in watts.

The Energy Analytics view provides a line chart with two series: consumed energy and saved energy. The Activity Log provides an audit trail of room changes and automated actions, while Smart Insights produces human-readable explanations such as prolonged light runtime, no detected person, sufficient ambient light, or energy savings relative to continuous operation.

The Settings view exposes the electricity tariff, automatic mode, dark mode, notifications, and reset controls. This configuration layer is important for experimentation because the same simulation can be rerun under different assumptions without changing source code.

VIII. ROOM MONITORING AND APPLIANCE MODEL

Fig. 4. Room monitoring concept for four simulated spaces

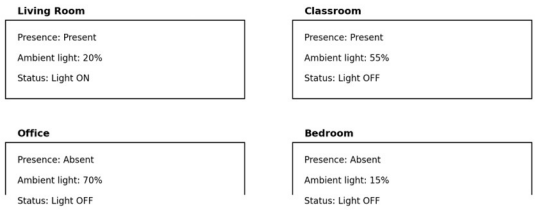


Fig. 4. Room monitoring concept for four simulated spaces.

The prototype initializes four rooms with different conditions. The Living Room begins with presence detected and a dark ambient-light value. The Classroom begins occupied with normal light. The Office begins unoccupied with a normal-to-bright boundary value, while the Bedroom begins unoccupied and dark. These different starting points allow the application to demonstrate more than one control decision immediately.

Each room is represented by a consistent state structure. The common structure is important because it allows new rooms to be added without inventing a new data model for each space. A room can therefore be treated as a reusable logical unit with the same sensing, appliance, and energy properties.

The lighting appliance has a power rating of 20 W. The fan has a power rating of 60 W. When a light is ON for one hour, its idealised simulated energy use is 0.020 kWh. When a fan is ON for one hour, its simulated energy use is 0.060 kWh. These are model parameters, not electrical measurements. Their purpose is to provide a consistent basis for demonstrating the energy formulas.

The current fan policy is manual. This is useful from a research perspective because it separates the automatically controlled load from a user-controlled load. A future experiment can therefore investigate the incremental benefit of adding automatic fan control, perhaps using occupancy and temperature as inputs.

The Room Monitor design also supports heterogeneous environmental states. A room can be occupied while bright, occupied while dark, unoccupied while dark, or unoccupied while bright. These combinations form a small state space that is sufficient for validating the basic logic and for building a complete test matrix.

Room	Initial presence	Initial light	Primary expected decision
Living Room	Present	20% (Dark)	Light ON
Classroom	Present	55% (Normal)	Light OFF
Office	Absent	70% (Normal boundary)	Light OFF
Bedroom	Absent	15% (Dark)	Light OFF

TABLE V. INITIAL ROOM CONDITIONS IN THE PROTOTYPE

IX. MATHEMATICAL MODEL

The energy model is deliberately simple so that every displayed value can be traced back to a small number of parameters. If an appliance has a power rating P in watts and is active for t hours, its energy consumption E in kilowatt-hours is defined by (1).

$$E = (P \times t) / 1000 \quad (1)$$

where P is appliance power in watts and t is operating time in hours.

For a room with n active appliances, total room energy is the sum of the individual appliance values:

$$E_{room} = \Sigma E_i \quad (2)$$

For the complete simulated building, total consumption is:

$$E_{total} = \Sigma E_{room,j} \quad (3)$$

The electricity cost is calculated using the configurable tariff C:

$$Cost = E_{total} \times C \quad (4)$$

The project compares controlled operation against a baseline in which the light and fan remain active for the elapsed time. Baseline energy is therefore:

$$E_{baseline} = ((P_{light} + P_{fan}) \times t) / 1000 \times N_{rooms} \quad (5)$$

The energy saving and saving percentage are calculated by:

$$E_{saved} = \max(0, E_{baseline} - E_{controlled}) \quad (6)$$

$$Saving\ (\%) = (E_{saved} / E_{baseline}) \times 100 \quad (7)$$

The max operator prevents a negative saving from being displayed when controlled operation happens to exceed the selected baseline. The formulas make an important methodological point: energy savings in this prototype are computed estimates generated by a model, not meter readings. If physical hardware is added later, the same formula structure can be retained while replacing estimated runtime and power with measurements from a smart meter or current sensor.

Parameter	Symbol	Prototype value	Unit
Light power	P_{light}	20	W
Fan power	P_{fan}	60	W
Electricity tariff	C	8	Rs/kWh
Simulation step	Δt	1	simulated minute
Rooms	N_{rooms}	4	rooms
Ambient-light range	L	0–100	percent

TABLE VI. MATHEMATICAL MODEL PARAMETERS

X. SIMULATION METHODOLOGY

The simulation methodology is based on repeated state updates. At the beginning of a session, the application loads a default or persisted state. Each room then has a known presence condition, ambient-light value, appliance state, and accumulated runtime. When the timer advances, the simulation increments elapsed time and evaluates each room.

For background rooms, the current prototype introduces small random changes in presence and ambient-light values. This produces a more dynamic Room Monitor during a live demonstration. The current room is controlled interactively so that the user can create deterministic cases by directly changing the presence button or light slider.

During every simulation tick, the automation rule is applied before energy is accumulated. If the light is ON, its runtime increases by one simulated minute and its energy contribution is calculated from the 20 W rating. If the fan is ON, its runtime increases and its energy contribution is calculated from the 60 W rating. Global totals are then recomputed from all room states.

Every three simulated minutes, the chart receives a new data point. The chart stores a bounded number of recent samples so that the visualization remains compact. This implementation choice is practical for a browser demo because it provides a moving trend without allowing unbounded growth of the chart arrays.

Evaluation in this paper uses two complementary approaches. First, the code behaviour is analysed directly using the

ISSN: 3107-6513

implemented rule and parameters. Second, controlled analytical scenarios are calculated for a fixed 60-minute period. These scenarios are not presented as field measurements. They are used to show what the implemented model implies under specific, explicitly stated conditions.

An eventual hardware study would extend the methodology by repeating the same scenarios with real sensors and measured loads. It would then compare the simulated estimates with physical observations, quantify sensor error, measure control latency, and evaluate whether occupants accept the switching behaviour.

Phase	Operation	Output
1	Initialise or load state	Room and global configuration
2	Read simulated inputs	Presence and light values
3	Classify light	Dark / Normal / Bright
4	Apply automation	Light ON or OFF
5	Accumulate runtime	Minutes per appliance
6	Calculate energy	kWh per room and global
7	Calculate baseline	Continuous-use reference
8	Update analytics	Chart and metrics
9	Record events	Activity log

TABLE VII. SIMULATION EXECUTION PIPELINE

XI. SCENARIO DESIGN AND FUNCTIONAL TESTING

Fig. 9. Functional test matrix for the implemented decision rule

ID	Presence	Light	Expected	Status
T1	Present	Dark	ON	Pass
T2	Present	Normal	OFF	Pass
T3	Present	Bright	OFF	Pass
T4	Absent	Dark	OFF	Pass
T5	Absent	Bright	OFF	Pass
T6	Auto mode OFF	Any	Manual	Pass

Fig. 9. Functional test matrix for the implemented decision rule.

A compact scenario matrix is sufficient to validate the principal light-control rule because the controller depends on two binary or categorical inputs: presence and ambient-light class. The minimum functional cases therefore cover occupied-dark, occupied-normal, occupied-bright, unoccupied-dark, and unoccupied-bright states. An additional case checks behaviour when automatic mode is disabled.

Test T1 verifies the positive activation condition. With a person present and ambient light classified as dark, the controller should turn the light ON. Test T2 and T3 verify that occupancy alone does not cause lighting: when the room is normal or bright, the light should remain OFF. T4 and T5 verify that absence takes priority and keeps the light OFF regardless of ambient-light level.

Test T6 verifies the system boundary between automatic and manual modes. When automatic mode is disabled, the decision message should indicate that the automatic rule is inactive. This provides a useful safeguard because it prevents the system from silently overriding a user who has intentionally selected manual control.

The application also supports secondary tests. Room selection should update the current room and append an activity log entry. Changing the tariff should update cost calculations without changing energy consumption. Enabling or disabling notifications should affect the toast behaviour but should not change the underlying control decision. Reset should clear accumulated energy, logs, and settings.

These tests are functional and behavioural. They do not prove physical sensor accuracy or electrical safety. A future hardware validation plan should add sensor calibration tests, false occupancy tests, response time measurement, relay switching reliability, power-meter comparison, and recovery testing after communication loss.

Test ID	Input state	Expected output	Purpose
T1	Present + Dark	Light ON	Activation condition
T2	Present + Normal	Light OFF	Daylight-aware control
T3	Present + Bright	Light OFF	Bright-room control
T4	Absent + Dark	Light OFF	Occupancy override
T5	Absent + Bright	Light OFF	Occupancy override
T6	Auto OFF	Manual mode message	Mode boundary
T7	Room switch	Room label and log update	Navigation integrity
T8	Tariff change	Cost changes, kWh unchanged	Calculation separation
T9	Notification OFF	No toast, state unchanged	UI preference
T10	Reset	State returns to defaults	Persistence integrity

TABLE VIII. FUNCTIONAL TEST CASES

XII. ANALYTICAL RESULTS AND DISCUSSION

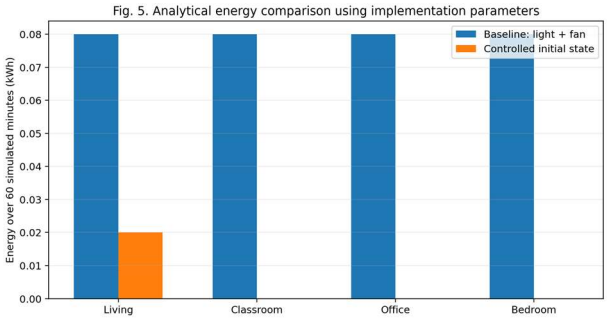


Fig. 5. Analytical energy comparison using implementation parameters. Values are calculated from the model, not field measurements.

To demonstrate the behaviour of the mathematical model, a controlled 60-minute analytical scenario can be calculated using the project parameters. The baseline assumes that both a 20 W light and a 60 W fan would run continuously. For one room, that baseline is 80 W for one hour, or 0.080 kWh. If the control rule keeps both devices OFF, controlled energy is zero. If only the light remains ON for the full hour, controlled energy is 0.020 kWh.

The initial state of the prototype provides a useful example. The Living Room starts occupied and dark, so the automatic light rule activates the light. The Classroom starts occupied but normal, so the light remains OFF. The Office and Bedroom start unoccupied, so their lights remain OFF. Under a fixed one-hour analytical interpretation without background drift, the four-room system therefore consumes 0.020 kWh of controlled lighting energy versus a 0.320 kWh continuous light-plus-fan baseline.

The corresponding analytical saving is 0.300 kWh, or 93.75 percent relative to that selected baseline. At the default tariff of Rs 8 per kWh, the estimated cost reduction is Rs 2.40 for that one-hour scenario. These values are useful for demonstrating the model but must not be interpreted as measured savings from the real world.

The scenario analysis also highlights an important modelling issue: the baseline definition strongly affects the reported percentage. If the baseline included only lighting rather than both light and fan, the percentage would be different. A publishable hardware evaluation should therefore define the baseline carefully and keep it consistent across trials.

The analytical result nevertheless supports the core hypothesis of the prototype: contextual control can reduce modelled runtime when the system avoids operating loads that are not required by the room state. The strength of the result lies in the transparent relationship between the input conditions, the control rule, and the resulting energy estimate.

Scenario	Baseline kWh / 60 min	Controlled kWh / 60 min	Saving kWh	Saving %	Interpretation
Present + Dark	0.080	0.020	0.060	75.0	Light required
Present + Normal	0.080	0.000	0.080	100.0	Ambient light sufficient
Present + Bright	0.080	0.000	0.080	100.0	Artificial light unnecessary
Absent + Dark	0.080	0.000	0.080	100.0	No occupant
Absent + Bright	0.080	0.000	0.080	100.0	No occupant + daylight

TABLE IX. ANALYTICAL 60-MINUTE SCENARIO RESULTS

ISSN: 3107-6513

XIII. ENERGY, COST AND TREND ANALYSIS

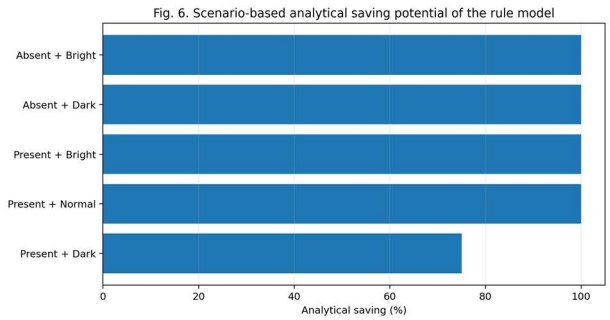


Fig. 6. Scenario-based analytical saving potential of the rule model.

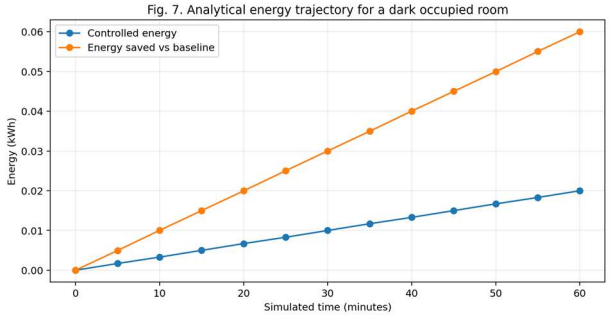


Fig. 7. Analytical energy trajectory for a dark occupied room.

The analytical saving percentages demonstrate the sensitivity of the model to context. The only scenario in the simplified matrix that retains lighting energy is Present + Dark because the rule deliberately treats that state as a genuine need for artificial illumination. All other tested combinations keep the light OFF. This is a direct consequence of the threshold logic rather than a learned prediction.

The trend model for a dark occupied room provides another perspective. If the light remains ON for 60 simulated minutes, the controlled energy increases linearly from zero to 0.020 kWh. A continuous light-plus-fan baseline increases more rapidly because the combined load is 80 W. The gap between the lines represents the simulated avoided energy under the selected baseline.

Cost follows the same relationship because the default tariff is constant. The project therefore treats energy reduction and cost reduction as two views of the same underlying consumption model. Changing the tariff changes the monetary output while leaving the physical energy estimate unchanged. This separation is desirable because tariffs can vary independently from appliance behaviour.

The analytics interface can expose these values over time rather than showing only a final number. This makes it possible to identify sudden changes, compare rooms, and explain why savings increase. The existing chart keeps the consumed and saved series together, which supports visual interpretation during a demonstration.

For a future research study, additional metrics can be added: average power, peak power, runtime utilisation, room-level saving percentage, and avoided operating minutes. These measures can help distinguish energy efficiency from simple device inactivity. The current prototype provides the data structures needed for such extensions.

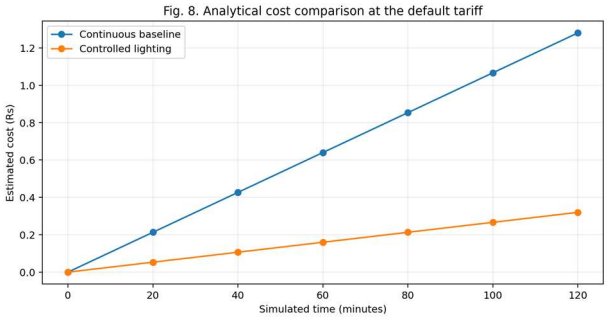


Fig. 8. Analytical cost comparison at the default tariff.

Tariff (Rs/kWh)	Baseline cost / 120 min	Controlled cost / 120 min	Analytical cost saved
4	0.64	0.16	0.48
8	1.28	0.32	0.96
10	1.60	0.40	1.20
12	1.92	0.48	1.44

TABLE XIV. TARIFF SENSITIVITY FOR THE ANALYTICAL 120-MINUTE CASE

For the 120-minute analytical example, the continuous baseline corresponds to 0.160 kWh and an estimated cost of Rs 1.28 at Rs 8 per kWh, while the controlled lighting case corresponds to 0.040 kWh and Rs 0.32. The difference, Rs 0.96, represents the modelled cost avoidance for that selected scenario.

Tariff sensitivity is linear in the current model. For a fixed energy-saving amount, doubling the tariff doubles the estimated monetary saving, while the energy saving in kWh remains unchanged. This separation is useful because it prevents electricity-price assumptions from being confused with the physical operating behaviour of the controller.

XIV. ACTIVITY LOGGING, SMART INSIGHTS AND OBSERVABILITY

An energy-control system should not only act; it should explain what it did. The prototype therefore includes an activity log that records events with a timestamp and a human-readable message. Room selection events, automatic light changes, and selected user actions can appear in this history. The log is capped so that the interface remains manageable during a long simulation.

The Smart Insights module provides a second form of observability. Instead of listing raw sensor values only, it interprets the current state. For example, it can report that a light has been running for many simulated minutes, that no person is detected, that ambient light is bright enough to make artificial lighting unnecessary, or that the system has saved energy relative to continuous operation.

This design supports explainability. A user can inspect the sensor state, read the decision message, check the appliance state, and then see the corresponding energy effect. Such a chain is valuable in educational demonstrations and in future field deployments because users may otherwise perceive an automatic system as arbitrary.

Notifications are also separated from the underlying state. The user can disable toast messages while the simulation continues to record the state and log events. This separation illustrates a good software design principle: presentation preferences should not silently change the control model.

Observability also supports debugging. When a control rule appears incorrect, the operator can review the room condition, decision text, and activity history. In a hardware version, the same principle can be expanded to include raw sensor values, timestamped communication status, actuator acknowledgement, and alarm conditions.

Feature	User-visible evidence	Engineering value
Decision message	Plain-language control reason	Explainability
Activity log	Time-stamped events	Auditability
Smart insights	Contextual observations	Decision support
Live power	Current watt estimate	Immediate state visibility
Chart	Consumed vs saved trend	Temporal analysis
Notifications	Action feedback	Fast awareness
Settings	Tariff and mode controls	Experiment configurability

ISSN: 3107-6513

TABLE X. OBSERVABILITY FEATURES

Time	Room	Trigger	Action	Explanation
10:01	Living Room	Present + Dark	Light ON	Artificial light required
10:05	Living Room	Light slider changed	Re-evaluate	Decision refreshed
10:08	Living Room	Presence absent	Light OFF	No occupant detected
10:12	Classroom	Room selected	Switch context	Operator monitoring room
10:15	Office	Bright condition	Light OFF	Ambient light sufficient

TABLE XV. EXAMPLE ACTIVITY-LOG EVENT MODEL

A useful extension is to treat the activity log as an event stream. Each event can be associated with a time, room, trigger, action, and explanation. Such a structure makes later export to a database or audit system straightforward.

XV. VALIDATION STRATEGY, LIMITATIONS AND THREATS TO VALIDITY

The current validation strategy is primarily software-level. It verifies that the state machine responds to combinations of presence and ambient-light inputs, that energy values are updated when appliances are active, that analytics remain consistent with accumulated state, and that the interface exposes expected information. This is appropriate for the current maturity level of the prototype.

The first limitation is the absence of physical sensor measurements. A simulated presence value cannot reveal false positives, false negatives, sensor blind spots, or motion-detection delays. Similarly, an LDR value represented as a percentage does not establish a real illuminance reading in lux. Therefore, the reported percentages in the simulation should not be interpreted as calibrated lighting measurements.

The second limitation is the simplified appliance model. Real devices may have different power draw, start-up behaviour, standby consumption, power-factor characteristics, and time-varying loads. The current light and fan values are fixed parameters intended for demonstration.

The third limitation is the baseline. The project compares controlled energy with a continuous-use reference. While this creates a clear demonstration of avoided runtime, a field study should use a carefully designed baseline based on observed or standard operating schedules. Different baseline definitions will produce different saving percentages.

The fourth limitation concerns occupant comfort. A controller that turns lights off too aggressively can save energy but reduce user satisfaction. The present rule uses fixed thresholds and does not learn occupant preferences. Future studies should investigate false-off events, override frequency, response delays, and acceptable minimum illumination levels.

The fifth limitation is security and deployment. The current application runs locally in a browser and does not expose network APIs. A future connected system would require authentication, secure communication, access control, firmware updates, logging integrity, and failure recovery.

Threat	Effect on conclusions	Mitigation for future study
Sensor error	Wrong presence/light state	Calibrate and repeat measurements
Simplified load model	Energy estimate may differ from reality	Install smart meter and record real load
Baseline choice	Saving percentage may change	Predefine baseline and report it explicitly
Fixed thresholds	Comfort may be inconsistent	Adaptive/user-centred thresholds
Background randomness	Non-repeatable demo states	Add deterministic test mode
Browser-only storage	No multi-user history	Use authenticated backend or local database

TABLE XI. THREATS TO VALIDITY AND MITIGATION

Validation level	Main question	Evidence	Status in current project
Level 1	Does the rule behave correctly?	Functional test matrix	Completed
Level 2	Are calculations internally consistent?	Analytical scenarios	Completed
Level 3	Do simulated inputs match sensors?	Sensor calibration data	Future
Level 4	Do measured loads match estimates?	Smart-meter traces	Future
Level 5	Does the system work for occupants?	Comfort and override study	Future

TABLE XVI. VALIDATION MATURITY LADDER

Validation should progress from deterministic software checks to physical measurements. Each level answers a different research question, so passing a software test should not be interpreted as evidence of sensor accuracy or field energy savings.

XVI. SCALABILITY AND MIGRATION TO HARDWARE

Fig. 10. Possible transition from software simulation to hardware deployment

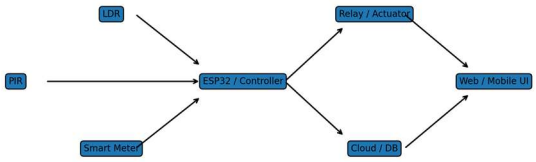


Fig. 10. Possible transition from software simulation to hardware deployment.

The prototype has been designed so that a later hardware implementation can preserve the conceptual architecture. The simulated presence input can be replaced by a physical PIR sensor connected to an ESP32 or similar microcontroller. The ambient-light slider can be replaced by an LDR circuit with an analogue reading. A relay or suitable solid-state actuator can provide the interface to the lighting load under appropriate electrical-safety requirements.

The energy model can also evolve. Instead of assuming fixed power ratings, a future system can measure actual current and voltage or obtain energy readings from a smart meter. The control engine can then use measured consumption to provide more accurate cost and saving estimates.

Communication can be added gradually. A local microcontroller can send sensor and appliance data to a web service over Wi-Fi. A backend database can then store long-term historical data, while the web interface can provide multi-day, weekly, or monthly analysis. The current modular rendering and state model provide a conceptual boundary between the control logic and the user interface.

Scalability also requires a more formal device model. Each appliance should have an identifier, nominal power, control type, current state, measurement source, and health state. Rooms should maintain references to the devices they contain. This would support buildings with dozens or hundreds of devices without duplicating interface logic.

Hardware migration should be treated as an engineering project rather than a direct copy of the simulation. Electrical isolation, relay ratings, fuse protection, grounding, enclosure design, sensor placement, power supply design, firmware reliability, and safe failure modes must be validated separately. The current paper intentionally does not claim that the browser simulation is itself a ready-to-connect mains controller.

Simulation component	Future physical equivalent	Additional requirement
Presence Boolean	PIR / occupancy sensor	Calibration and placement
0–100 light value	LDR + ADC	Voltage-to-light mapping
20 W light model	LED/lighting circuit	Relay/actuator safety
60 W fan model	Fan + rated actuator	Electrical isolation
Local storage	Database / cloud storage	Authentication and backup
Chart arrays	Time-series data store	Sampling and retention policy

TABLE XII. SOFTWARE-TO-HARDWARE MAPPING

XVII. SECURITY, RELIABILITY, USER EXPERIENCE AND FUTURE WORK

Although the present prototype is local and low risk, a connected smart-building implementation would introduce security requirements. Sensor data can reveal occupancy patterns, while control interfaces can affect physical appliances. A future deployment should therefore use authenticated access, encrypted communication, least-privilege

ISSN: 3107-6513

device permissions, and tamper-resistant logs. Device credentials should not be embedded in browser code.

Reliability is equally important. A smart controller should define what happens when a sensor stops responding, a Wi-Fi connection is lost, a device does not acknowledge a command, or a software process restarts. Safe default states and timeout behaviour should be documented before physical deployment. For lighting, a temporary loss of automation might reasonably revert to manual control rather than continuously cycling the load.

User experience should remain a design requirement. Automatic switching can be annoying when thresholds are poorly chosen, and users may override the system. A future version can record manual overrides and use them to tune thresholds or to identify rooms where the fixed rule is not suitable. This would connect the simple deterministic prototype to occupant-centred approaches reported in the literature [2].

Potential future research directions include adaptive threshold learning, occupancy prediction, temperature-aware fan control, real-time metering, carbon-intensity-aware scheduling, demand response, mobile notifications, multi-user roles, and integration with renewable generation or battery storage. Another useful extension is a reproducible experiment mode in which the random background behaviour is disabled and a predefined sequence of sensor states is replayed.

A larger evaluation can also compare the proposed rule with alternative strategies: manual control, occupancy-only control, light-only control, occupancy-plus-light control, and adaptive control. Such a comparison would isolate the contribution of combining the two contextual inputs. The same experimental framework could then quantify energy, comfort, response time, and override rate.

Overall, the future work is best viewed as a staged pathway: deterministic software simulation, hardware-in-the-loop validation, controlled laboratory measurement, pilot deployment, and long-term field study. Each stage reduces a different category of uncertainty.

Future feature	Benefit	Suggested evaluation
Adaptive light threshold	Better personalisation	Override and comfort study
Temperature-aware fan control	Broader appliance savings	Energy + comfort comparison
Real-time meter	Measured energy	Meter-vs-model error
Cloud history	Long-term analytics	Data retention and latency
Demand response	Grid-aware operation	Peak-period load shift
Occupancy prediction	Proactive control	Prediction accuracy and savings
Mobile alerts	Remote awareness	Usability and alert acceptance

TABLE XIII. PRIORITISED FUTURE EXTENSIONS

XVIII. CONCLUSION

This paper presented a software simulation for a Smart Energy Saving System that combines simulated presence detection and ambient-light analysis to control lighting in multiple rooms. The project demonstrates how a small set of transparent rules can be connected to energy calculation, cost estimation, monitoring, logging, analytics, and user-facing insights.

The most important architectural decision is the separation between inputs, decision logic, appliance state, and presentation. This allows the prototype to remain understandable while also creating a pathway toward future hardware. The current browser implementation can therefore be viewed as a digital prototype of a building-control concept rather than as a substitute for physical validation.

The mathematical model is intentionally simple: energy depends on power and runtime, cost depends on energy and tariff, and saving is calculated relative to a clearly stated baseline. This transparency supports reproducible scenario analysis. The analytical cases show that the model can reduce simulated energy when lighting is avoided in rooms that are bright or unoccupied. However, the paper deliberately distinguishes analytical calculations from measured field performance.

The prototype also illustrates the importance of observability in automation. Decision messages, activity logs, smart insights, room cards, live power values, and trend charts allow the user to understand not only what the system is doing but why it is doing it. This characteristic is particularly useful when presenting the project to evaluators, students, researchers, or potential hardware developers.

The next research step is therefore not simply to add more interface features, but to validate the same control logic against real sensors and measured electrical consumption. Physical tests should quantify sensor accuracy, switching latency, false-off events, energy use, occupant acceptance, and reliability. If those results are favourable, the architecture can be extended toward networked smart-building management.

ACKNOWLEDGMENT

The author thanks the academic and project-development environment that supported the design and implementation of the Smart Energy Saving System prototype. The project was developed as a software simulation to explore intelligent energy-control concepts before physical hardware deployment.

REFERENCES

[1] C. de Bakker, M. Aries, H. Kort, and A. Rosemann, “Occupancy-based lighting control in open-plan office spaces: A state-of-the-art review,” Building and Environment, vol. 112, pp. 308–321, 2017, doi: 10.1016/j.buildenv.2016.11.042.

[2] Z. Nagy, F. Y. Yong, M. Frei, and A. Schlueter, “Occupant centered lighting control for comfort and energy efficient building operation,” Energy and Buildings, vol. 94, pp. 100–108, 2015, doi: 10.1016/j.enbuild.2015.02.053.

[3] T. Labeodan, C. de Bakker, A. Rosemann, and W. Zeiler, “On the application of wireless sensors and actuators network in existing buildings for occupancy detection and occupancy-driven lighting control,” Energy and Buildings, vol. 127, pp. 75–83, 2016, doi: 10.1016/j.enbuild.2016.05.077.

[4] D. Caicedo, A. Pandharipande, and G. Leenders, “A review on lighting control technologies in commercial buildings, their performance and affecting factors,” Renewable and Sustainable Energy Reviews, vol. 33, pp. 268–279, 2014, doi: 10.1016/j.rser.2014.01.090.

[5] Y. Chu, P. Guillante, D. Mitra, R. Mahmud, and K. Cetin, “Typical academic building energy model development and energy saving evaluation using occupant-based control,” Journal of Building Engineering, vol. 79, 107818, 2023, doi: 10.1016/j.jobe.2023.107818.

[6] V. Balasubramaniam et al., “IoT Considerations, Requirements, and Architectures for Smart Buildings—Energy Optimization and Next-Generation Building Management Systems,” IEEE Internet of Things Journal, vol. 4, no. 1, pp. 269–283, 2017, doi: 10.1109/JIOT.2017.2647881.

[7] International Energy Agency, Energy Efficiency 2024, IEA, Paris, 2024.

[8] L. Gao, Y. Yuan, L. Xiao, W. Li, J. Qin, J. Wu, and Y. Chen, “Occupant-based control of lighting system for multi-person office rooms based on WiFi probe technology,” Building and Environment, vol. 269, 112421, 2025, doi: 10.1016/j.buildenv.2024.112421.

[9] Z. Nagy, F. Y. Yong, and A. Schlueter, “Occupant centered lighting control: A user study on balancing comfort, acceptance, and energy consumption,” Energy and Buildings, vol. 126, pp. 310–322, 2016, doi: 10.1016/j.enbuild.2016.05.075.

[10] L. Swarup, R. R. Panda, J. Singh, R. Savitha, D. N. S. R. Kumar, and D. K. Yadav, “Integration of IoT and Sensor

ISSN: **3107-6513**

Technology for Smart Energy Management in Buildings," in Proc. 2025 International Conference on Networks and Cryptology (NETCRYPT), pp. 336–341, 2025, doi: 10.1109/NETCRYPT65877.2025.11102500.

[11] International Journal of Advanced Multidisciplinary Research and Educational Development, "Paper preparation template," IJAMRED, ISSN 3107-6513.

[12] Chart.js Documentation, "Open source JavaScript charting library," Chart.js project documentation.

AUTHOR NOTE

Project implementation basis: the accompanying Smart Energy Saving System browser application defines four rooms, 20 W lighting, 60 W fan load, a default tariff of Rs 8 per kWh, simulated presence and ambient-light inputs, runtime-based energy calculations, room monitoring, analytics, activity logging, and smart insights. The prototype is explicitly a simulation and does not claim direct ESP32, PIR, or LDR hardware measurements.