WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

VOLTFLOW: AI HOUSE ROUTING AND UTILITY PLANNING SYSTEM

Dr. K. N. Tayade[1], Pranav Alaspure[2], Om Bhomle[3], Soham Chawlewar[4]

[1] Department of Information Technology, Government College of Engineering, Amravati.

[2–4] B.Tech Students, Department of Information Technology, Government College of Engineering, Amravati.

ABSTRACT :

VoltFlow: AI House Routing and Utility Planning System is a web-based application that automatically designs the electrical wiring and plumbing of a house from an ordinary floor-plan image. In most Indian homes these services are decided on site without a drawing, which leads to wasted material, repeated purchases and code violations that are discovered only after plastering. VoltFlow addresses this by combining computer vision, rule-based engineering design and graph-based routing in a single browser application. A Python server (FastAPI, OpenCV, EasyOCR) detects walls, openings and named rooms; the browser then sizes lights, fans, sockets and switchboards to each room, routes wires inside the walls and ceilings and pipes to inspection chambers, renders the result in 3D, and produces a bill of materials with a cost estimate. The design follows IS 732:2019 for wiring [1] and NBC 2016 / SP 35 for plumbing [2]. On a real two-bedroom drawing, the system recovered the scale exactly (19.38 mm per pixel), found and named all 7 rooms, produced 48 electrical points on 10 circuits, and placed no wire segment outside the building.

Keywords: VoltFlow, Floor-Plan Analysis, Computer Vision, Electrical Wiring Design, Plumbing Auto-Routing, IS 732:2019, Bill of Materials, 3D Visualization, Building Services (MEP).

1. INTRODUCTION

In most Indian houses, the wiring and plumbing are decided on site by the electrician or plumber, without any drawing. Material is therefore bought by guesswork, which results either in waste or in many extra trips to the shop. Errors such as a missing earth wire, a wrong wire size for an air conditioner or a socket placed inside a bathroom are usually found only after the walls have been plastered [1]. Professional MEP (Mechanical, Electrical and Plumbing) design packages such as Revit MEP and AutoCAD MEP solve this problem, but they are costly and need trained engineers, so they are out of reach for ordinary home builders and small contractors.

This paper presents VoltFlow, a browser-based system that plans the electrical and plumbing services of a house automatically. The user uploads a picture of a floor plan; the software finds the walls, doors, windows and rooms, builds a 3D model, places lights, fans, sockets and switchboards in every room, draws the wire and pipe routes, and finally reports how many metres of each wire size, how many switches, lights and fans are needed, and the approximate cost in rupees. The design rules follow the Indian standards IS 732:2019 for wiring [1] and NBC 2016 / SP 35 for plumbing [2]. The main contributions of this work are:

- A computer-vision pipeline that converts a 2D floor-plan image into wall geometry with real dimensions in millimetres, detects doors, windows and closed rooms, and reads room names by OCR [8].
- An area-based electrical auto-planner that sizes lights, fans, sockets, switchboards, circuits and the distribution board to each room according to IS 732:2019 [1].
- A routing engine that keeps all wires inside the walls and ceilings of the house using shortest-path and Steiner-tree search [9], [10], together with an automatic plumbing router for supply, drainage and inspection chambers.
- An interactive 3D viewer [12] and a bill-of-materials calculator that converts the planned points and routes into wire lengths, item counts and a cost estimate using averaged Indian market prices.

The rest of the paper is organized as follows. Section 2 reviews the related work, Section 3 describes the proposed system and its eight-step pipeline, Section 4 describes the implementation and technology stack, Section 5 presents and discusses the results, and Section 6 gives the conclusion and future scope.

2. LITERATURE SURVEY

Building-services design in India is governed by published codes. IS 732:2019 specifies the wire size permitted for each breaker rating, the maximum number of points on a lighting circuit, earth-leakage protection for sockets and wet areas, and the use of separate circuits for heavy loads such as air conditioners and water heaters [1]. Plumbing practice follows NBC 2016 Part 9 and SP 35, which require gravity supply from an overhead tank, separate

soil and waste lines, traps, inspection chambers at bends and a fall toward the outlet [2]. Single-core flame-retardant copper house wire is supplied to IS 694 in standard 90 m coils [3]. These documents are written for engineers; there is little tooling that applies them automatically to a given house plan.

Recovering walls and rooms from a drawing is a classical image-analysis task. Binarisation separates ink from paper, and the distance transform gives the thickness of every line, which allows thick walls to be told apart from thin furniture lines, text and dimension arrows [5]. Morphological opening with long horizontal and vertical structuring elements keeps only straight runs of a minimum length, so that symbols and lettering are removed while walls survive [6]. Closed white regions are then grouped by connected-component labelling, and their outlines are simplified with the Douglas–Peucker algorithm [7]. The OpenCV library provides optimised implementations of all of these operations [4]. VoltFlow combines them with real-world thresholds (for example 75 mm and 172 mm for wall thickness) so that results are expressed in millimetres rather than pixels.

Reading the room label printed in each room is done with optical character recognition. EasyOCR works offline and gives good accuracy on printed text [8], but small misreadings such as “Batnroom” are common on scanned drawings, so VoltFlow corrects the output by fuzzy matching against a vocabulary of room names. For wire and pipe routing, the walls of the house are treated as a graph. Dijkstra’s algorithm finds the cheapest path between two nodes of such a graph [9], and a Steiner-tree approximation connects many nodes with a single branching network that uses the least total length [10]. Using only wall segments as edges guarantees that a route cannot leave the building.

Finally, modern browsers can render interactive 3D scenes with WebGL through Three.js [12], and lightweight Python frameworks such as FastAPI make it simple to expose heavy image analysis as a REST service [11]. VoltFlow builds on these tools to deliver the complete chain from a floor-plan image to a priced bill of materials inside a single web application, which, to the best of our knowledge, is not offered by the commercial MEP packages in an automatic and low-cost form.

3. PROPOSED SYSTEM

3.1. System Overview and Pipeline

VoltFlow works in eight sequential steps in which the output of each step is the input of the next: (1) upload of the floor plan, (2) scale calibration, (3) computer vision on the server, (4) review and validation, (5) electrical auto-planning, (6) plumbing auto-routing, (7) 3D model generation, and (8) bill of materials and cost estimation. Steps 1–2 and 4–8 run in the web browser, while step 3 (image analysis) runs on the Python server, as shown in Fig. 1. This split keeps the interactive parts fast and lets the heavy image processing use the Python computer-vision ecosystem [4].

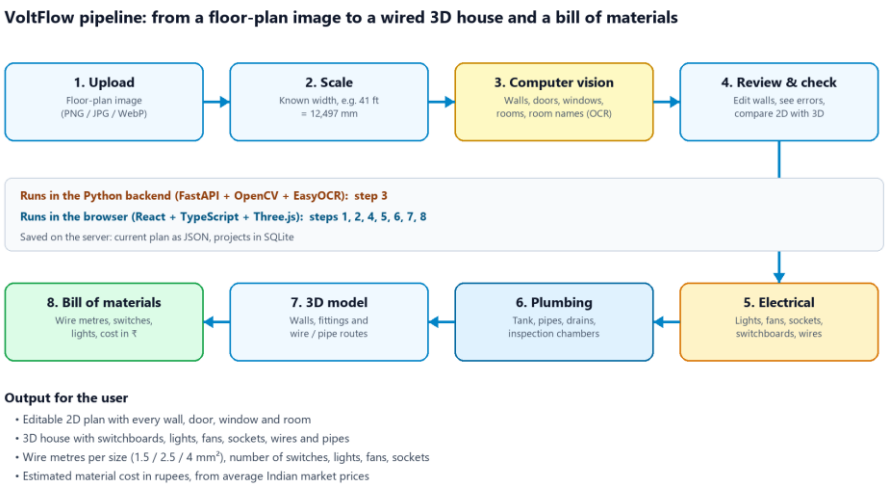


Fig. 1 – The complete VoltFlow pipeline, from floor-plan image to bill of materials.

3.2. Upload and Scale Calibration

The user uploads a PNG, JPG or WebP drawing and can rotate it if required; a ready-made Indian template such as a 2BHK flat can be used instead. A picture has pixels while a house has millimetres, so the user enters one known real length, usually the building width printed on the drawing (for example 41 ft = 12,497 mm). The server measures the same width in pixels between the outer faces of the walls and computes millimetres per pixel = known width ÷ pixel width. For the example drawing this gives 19.38 mm per pixel. Only long wall runs are measured, because dimension lines, arrows and text outside the house would otherwise distort the scale by 15–20%.

3.3. Computer Vision Pipeline

The image is sent to the FastAPI endpoint /api/ai/analyze-floorplan [11], where OpenCV [4] processes it in the following order:

- **Binarise:** the image is converted into black ink and white paper.
- **Separate walls from details:** a distance transform [5] measures the thickness of each line; lines thicker than 75 mm (real size) are walls, and thin lines are furniture, text and dimensions.
- **Find wall runs:** morphological opening with long horizontal and vertical kernels [6] keeps only straight runs longer than 700 mm.
- **Classify thickness:** runs of 172 mm or more are 230 mm (9″) external walls and thinner runs are 115 mm (4.5″) partitions, which is standard Indian brickwork.
- **Close doors and windows:** each wall run is extended along its own line until it meets a wall again, so that a doorway or window no longer leaks one room into the next.

- **Find rooms:** every closed white area inside the walls is a room (connected-component labelling); its outline, area and centre are measured and the outline is simplified [7].
- **Read room names:** EasyOCR [8] reads the text, misread words are corrected against a room vocabulary, two labels in one open space are kept together (e.g. "Living Room / Dining Room"), and duplicate names are numbered.
- **Detect openings:** gaps of 600–2,600 mm along a wall line are doors (inside) or windows (outside).
- **Return JSON:** walls with start and end points and thickness, rooms with polygons and names, openings and a validation report.

The browser then shows the detected walls over the drawing, where the user can delete a wrong wall or change its thickness. A geometric validator checks that the measured width equals the known width, that all wall ends are connected, that walls are straight (0° / 90°) and that openings lie inside their walls. A side-by-side 2D / 3D comparison is shown and the user confirms the result. Fig. 2 and Fig. 3 show an example input and the detected output.

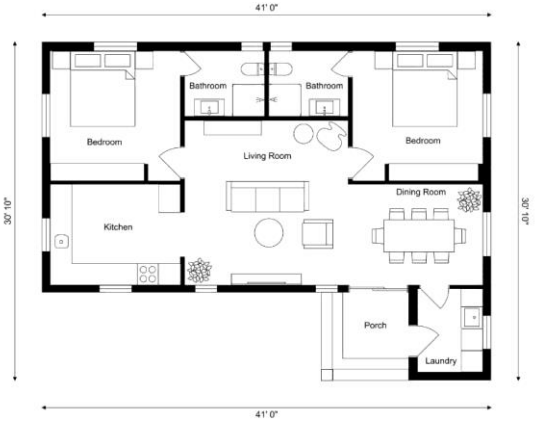


Fig. 2 – Example input: a two-bedroom plan, 41'0" × 30'10".



Fig. 3 – Result: 7 rooms found and named, 12 walls (red = 230 mm, blue = 115 mm).

3.4. Electrical Auto-Planning

For every room the planner decides what is needed from the room type and floor area (Table 1). Every room also receives one bulb in a lamp holder, and a switchboard is placed beside the room's door at a height of 1,200 mm. The switchboard holds one switch per light or fan and one regulator per fan, and its size (2 to 18 modules) is calculated. Sockets are allotted by use: bedrooms get an AC point and bedside sockets; the living room gets a TV point, sockets and an AC point; the kitchen gets counter, fridge and RO points; the bathroom gets a geyser point; and the laundry gets a washing-machine point.

Table 1. Area-based sizing of lighting and fans

Room area	Tube lights (20 W LED)	Ceiling fans (1200 mm)
Below 15 m²	1	1
15 – 28 m²	2	2 (from 16 m²)
28 – 42 m²	3	3 (from 30 m²)
Bathroom / toilet	0 (one wall bulb instead)	0 (exhaust fan instead)
Kitchen	By area	1 only if 12 m² or more, plus exhaust fan

Circuits follow IS 732:2019 [1]: lights and fans are on a 10 A MCB with 1.5 mm² wire and at most 10 points per circuit; 6 A sockets are on a 16 A RCBO with 2.5 mm² wire; 16 A power points are on a 20 A RCBO with 4 mm² wire; and every air conditioner and every geyser has its own circuit. The distribution board (DB) is placed in the living room near the entrance and its size is chosen from the number of breakers it must hold.

3.5. Wire Routing Inside the Building

Feeders run from the DB to each room switchboard along the centre lines of the walls. Because the path uses only walls of the house, it cannot leave the building. From each switchboard, every light, fan and socket gets its own run, as Indian electricians do in point wiring. Wall points (tube lights, sockets and exhaust fans) are reached along the room's own wall faces by the shorter way round, while ceiling points (fans and bulbs) are reached across that room's ceiling slab with a right-angled run; every run is tested point by point to stay inside the room outline. A Dijkstra shortest-path search [9] combined with a Steiner-tree approximation [10] connects all boards with the least wire and lets circuits share conduits. A voltage-drop check, drop % = 2 × length × current × copper resistivity ÷ (wire size × 230 V), warns when the drop exceeds 5%. The generated wiring for the example plan is shown in Fig. 4.



Fig. 4 – Generated wiring: blue squares are switchboards, the black square is the DB, circles are fans and coloured lines are circuits.

3.6. Plumbing Auto-Routing, 3D Model and Bill of Materials

Wet rooms (bathrooms, kitchen and utility) are identified from the room names. An overhead tank of 1000 L is placed above the plumbing shaft and a vertical down-take feeds a manifold; cold and hot CPVC lines run along the walls to each basin, shower, WC and sink, with a geyser feeding the hot line. Drain pipes (SWR 75 / 110 mm) leave each wet room through the wall to a trench around the house, and inspection chambers (IC-1 to IC-4) are placed at the corners, the main chamber being connected to the municipal sewer or a soak pit, in line with NBC 2016 [2].

The 3D model is drawn with Three.js through React Three Fiber [12]. Walls are drawn with real thickness and height together with the plinth, doors, windows and every fitting: switchboards with the correct number of switches and regulator knobs, tube lights, bulbs, fans, sockets, the DB and all wire and pipe routes. Wires are drawn slightly transparent on top of the walls so that hidden routes remain visible, and the user can rotate, zoom, cut the walls at 1 m or floor level, and switch on an X-ray view.

The bill-of-materials calculator adds the length of every route, the vertical drops to each switchboard and point, the tail at the DB and 15% wastage. Because Indian houses use single-core wires, each run needs three wires: phase (red), neutral (black) and earth (green), supplied in 90 m coils [3]. Item counts come directly from the planned points, and each item is priced using averaged Indian retail prices including GST.

4. SYSTEM IMPLEMENTATION AND TECHNOLOGY STACK

VoltFlow is implemented as a single web application with a browser front end and a Python back end (Fig. 5). The front end is written in TypeScript with React 19 and built with Vite 8. The 3D scene uses Three.js with React Three Fiber and Drei [12], and three-bvh-csg cuts door and window holes in the 3D walls. The 2D plan editor uses Konva (react-konva) for fast canvas drawing with drag and drop. Application state and undo / redo are handled by Zustand with zundo, Zod validates the plan file format so that broken plans cannot be loaded, Clipper2 performs polygon union and offset operations, MapLibre GL with proj4 provides the site view, i18next keeps all interface text in one place, and oxlint performs fast code-quality checks.

The back end runs on Python 3 with FastAPI and Uvicorn [11], exposing the REST endpoints /api/ai/analyze-floorplan, /api/plan.php and /api/projects. Image analysis uses OpenCV [4] and NumPy, EasyOCR on PyTorch [8] reads room names, Pillow decodes uncommon image formats, and SQLite stores saved projects and route results without needing a separate database server. The application is built with Node.js 24 and npm for the front end and pip for the back end, and is packaged as a Docker image (Node build with Apache/PHP static hosting) with GitHub Actions for continuous build checks. Testing consists of an engineering scenario suite (geometry accuracy, electrical, plumbing, L-shaped plans and save / load) and Python unit tests for the API and the vision engine.

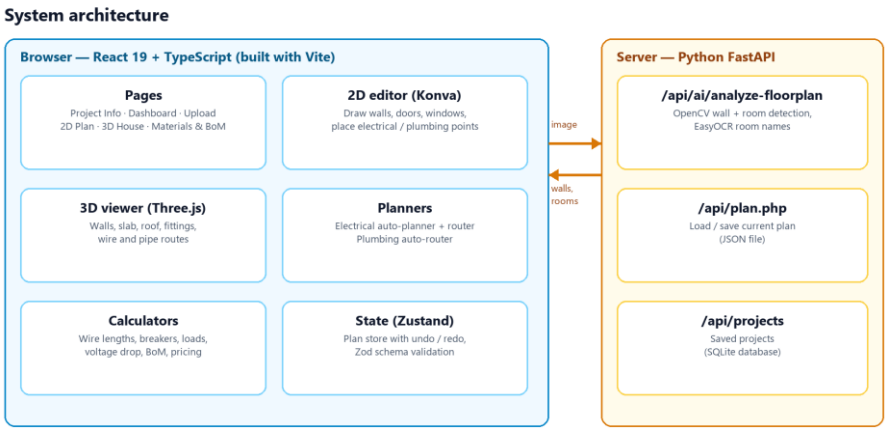


Fig. 5 – How the browser application and the Python server work together.

Table 2. Indian standards followed by VoltFlow

Standard	What is applied
IS 732:2019 – Electrical wiring installations [1]	Wire size per breaker (1.5 mm² ≤ 10 A, 2.5 mm² ≤ 16 A, 4 mm² ≤ 25 A); maximum 10 points per lighting circuit; 30 mA RCD for sockets and wet areas; separate circuits for heavy loads; earth wire everywhere
IS 694 – PVC insulated cables [3]	Single-core FR copper house wire in standard 90 m coils
NBC 2016 Part 9 / SP 35 – Plumbing [2]	Overhead-tank gravity supply, separate soil and waste lines, traps, inspection chambers at bends, falls toward the outlet
NBC 2016 Part 6 – Structure	230 mm external and 115 mm internal brick walls

5. RESULTS AND DISCUSSION

The system was evaluated on a real two-bedroom drawing measuring 41′0″ × 30′10″ (Fig. 2). The results are summarised in Table 3. Using the 41 ft building width as the known length, the server obtained a scale of 19.38 mm per pixel and a measured building width of 12,500 mm, which agrees with the declared width. All 7 closed rooms (2 bedrooms, 2 bathrooms, living / dining, kitchen and laundry) were found and named correctly. The 12 detected walls were fully connected, with no unconnected ends, and all were straight.

Table 3. Results on the example drawing

Check	Result
Scale from 41 ft width	19.38 mm/px; measured building width 12,500 mm (exact)
Rooms found and named	7 of 7 closed rooms: 2 bedrooms, 2 bathrooms, living / dining, kitchen, laundry
Walls	12 connected walls, 0 unconnected ends, 100% straight
Electrical points	48 points on 10 circuits, all connected
Wires outside the building	0
Automated tests	All engineering scenarios A–G and 15 Python tests pass

For the electrical design, the planner placed 48 points on 10 circuits, all of which are connected to a switchboard, and the routing check found 0 wire segments and 0 points outside the building (Fig. 4). The resulting bill of materials is given in Table 4. For this plan the wire requirement is 444 m of 1.5 mm² wire for lights and fans (148 m each of red, black and green, about 5 coils of 90 m), 366 m of 2.5 mm² wire for 6 A sockets and 393 m of 4 mm² wire for AC, geyser and 16 A points, each of the last two also about 5 coils. The total material cost is about ₹1,44,500.

Table 4. Bill of materials for the example 2BHK plan

Item	Quantity
1.5 mm² wire – lights and fans	444 m (148 m each red / black / green ≈ 5 coils of 90 m)
2.5 mm² wire – 6 A sockets	366 m (≈ 5 coils)
4 mm² wire – AC, geyser, 16 A points	393 m (≈ 5 coils)
6 A / 16 A / 20 A DP switches	31 / 5 / 5
Fan regulators	5
Switchboards	3 × 2-module, 1 × 3-module, 2 × 6-module, 1 × 16-module
LED tube lights / lamp holders + bulbs	6 / 7 + 7
Ceiling fans / exhaust fans	5 / 3
Sockets 6 A / 16 A / AC	10 / 5 / 3
DB and protection	24-module DB, 2-pole main, 4 RCBOs, 6 MCBs, voltage relay
Estimated material cost	≈ ₹1,44,500 (average Indian retail prices, incl. GST)

The prices used are averages of several current listings (Flipkart, Smartprix, IndiaMART, Justdial, the Philips India store and wire makers' 2026 price lists with the usual dealer discount), checked on 28 Oct 2026. They are a planning estimate and should be confirmed by a local quote, since copper rates change frequently.

The application exposes these results through seven screens: Project Info, Dashboard (summary of walls, rooms, electrical points and pipes with one-click auto-generation), Upload Plan (Upload → Scale → Review walls → Check → Compare and confirm), 2D Plan and Utilities (Structure, Electrical and Plumbing modes with auto-route buttons), 3D House (layers for walls, electrical, water, drainage, X-ray and cut-away), Materials and BoM Cost, and Settings (code standards, wall type, tank height and local material rates). Together they let a user move from a drawing to a priced plan without installing any software beyond a web browser.

Limitations. The vision engine works best on clean, high-contrast drawings with solid walls; hand sketches and photographs taken at an angle are harder. Open areas such as a porch with only railings are not counted as rooms, curved and slanted walls are not detected (most Indian plans are rectangular), only a single floor is supported, and prices are averages that move with copper rates.

6. CONCLUSION AND FUTURE SCOPE

This paper presented VoltFlow, an AI-assisted system that takes a plain floor-plan image all the way to a code-compliant wiring and plumbing plan, an interactive 3D model and a priced bill of materials, automatically and inside a web browser. By combining distance-transform and morphology-based wall detection OCR-based room naming area-based electrical sizing, shortest-path and Steiner-tree routing and rule checks from IS 732:2019 and NBC 2016 the system removes the guesswork that normally surrounds house wiring and plumbing. On a real two-bedroom drawing it recovered the scale exactly, named all 7 rooms, generated 48 connected electrical points on 10 circuits with no wire outside the building, and estimated the material cost at about ₹1,44,500.

In future work, deep-learning wall segmentation can be added to handle photographs and hand-drawn plans, and the system can be extended to multi-storey houses with staircases and vertical risers. Live prices from shop APIs and city-wise rates would make the cost estimate more accurate, and export to DXF / PDF drawings would give the electrician and plumber a ready-to-use document. Further extensions include solar-panel and inverter planning, smart-home wiring, and a mobile application with an augmented-reality view on the actual site.

REFERENCES :

1. Bureau of Indian Standards, “IS 732:2019 – Electrical wiring installations – Code of practice,” BIS, New Delhi, 2019.
2. Bureau of Indian Standards, “National Building Code of India 2016, Part 9: Plumbing Services,” and “SP 35: Handbook on Water Supply and Drainage,” BIS, New Delhi.
3. Bureau of Indian Standards, “IS 694:2010 – PVC insulated cables for working voltages up to and including 1100 V,” BIS, New Delhi, 2010.
4. G. Bradski, “The OpenCV Library,” Dr. Dobb’s Journal of Software Tools, 2000.
5. G. Borgefors, “Distance transformations in digital images,” Comput. Vis. Graph. Image Process., vol. 34, no. 3, pp. 344–371, 1986.
6. R. M. Haralick, S. R. Sternberg, and X. Zhuang, “Image analysis using mathematical morphology,” IEEE Trans. Pattern Anal. Mach. Intell., vol. PAMI-9, no. 4, pp. 532–550, Jul. 1987.
7. D. H. Douglas and T. K. Peucker, “Algorithms for the reduction of the number of points required to represent a digitized line or its caricature,” Cartographica, vol. 10, no. 2, pp. 112–122, 1973.
8. JaidedAI, “EasyOCR: Ready-to-use OCR with 80+ supported languages,” GitHub repository, 2020. [Online]. Available: https://github.com/JaidedAI/EasyOCR
9. E. W. Dijkstra, “A note on two problems in connexion with graphs,” Numerische Mathematik, vol. 1, no. 1, pp. 269–271, 1959.
10. L. Kou, G. Markowsky, and L. Berman, “A fast algorithm for Steiner trees,” Acta Informatica, vol. 15, no. 2, pp. 141–145, 1981.
11. S. Ramírez, “FastAPI: high performance, easy to learn, fast to code, ready for production,” documentation. [Online]. Available: https://fastapi.tiangolo.com
12. Three.js Authors, “Three.js – JavaScript 3D library,” documentation. [Online]. Available: https://threejs.org