# Project Architecture and Main Algorithm Explanation

## Project Architecture

The project is organized into several key components, each with a specific responsibility to ensure a clean and maintainable structure. The primary components are:

1. **HTML**:
   - Contains the structure of the building interface, defining the floors and space for elevators.
2. **CSS**:
   - Styles the floors, buttons, and elevators, ensuring a visually appealing and responsive design.
3. **JavaScript**:
   - Divided into multiple files for better organization and separation of concerns:
     - `Elevator.js`: Defines the `Elevator` class, representing each elevator in the building.
     - `Floor.js`: Defines the `Floor` class, representing each floor with its call button.
     - `main.js`: Handles the initialization of floors and elevators, and the main logic for the elevator movement.

### Elevator Class (`Elevator.js`)

The `Elevator` class encapsulates the properties and behaviors of an elevator, including its position, current floor, movement logic, and interaction with the DOM.

**Key Methods:**

- `constructor(id, leftPosition)`: Initializes the elevator with an ID and a position.
- `createElevator()`: Creates the HTML representation of the elevator.
- `moveToFloor(floorNumber)`: Handles the movement of the elevator to a specified floor, including smooth animations and sound effects.
- `appendTo(parent)`: Appends the elevator element to a parent container in the DOM.

### Floor Class (`Floor.js`)

The `Floor` class encapsulates the properties and behaviors of a floor, including its number, call button, and interaction with the DOM.

**Key Methods:**

- `constructor(number, requestElevatorCallback)`: Initializes the floor with a number and a callback function for requesting an elevator.
- `createFloor()`: Creates the HTML representation of the floor and its call button.
- `requestElevator()`: Handles the logic for requesting an elevator and changing the button color.
- `prependTo(parent)`: Prepends the floor element to a parent container in the DOM.

## Main Logic (`main.js`)

The main logic initializes the building with elevators and floors, and handles the elevator request and movement logic.

### Key Functions:

- `requestElevator(floorNumber)`: Determines the closest available elevator and moves it to the requested floor.

### Main Algorithm Explanation

The core algorithm for the elevator system focuses on efficiently handling elevator requests and moving the closest available elevator to the requested floor. Here's a step-by-step breakdown:

1. **Initialization**:
   - Create instances of `Elevator` and `Floor` classes.
   - Append these instances to the DOM, setting up the building structure.
2. **Elevator Request Handling**:
   - When a floor button is pressed, the `requestElevator` function is invoked.
   - The function iterates through all elevators to find the closest one that is not currently moving.
   - The distance between the elevator's current floor and the requested floor is calculated.
   - The closest elevator is selected and moved to the requested floor.
3. **Elevator Movement**:
   - The `moveToFloor` method of the `Elevator` class is called with the target floor number.
   - This method calculates the new position based on the floor height and animates the elevator's movement smoothly.
   - Once the elevator reaches the target floor, a sound is played, and the elevator pauses for 2 seconds before becoming available for new requests.
4. **Button Color Change**:
   - When a floor button is pressed, its color changes to green.
   - Once the elevator reaches the requested floor, the button color reverts to its original state.

This algorithm ensures that the elevators operate efficiently, responding to requests promptly and providing a visually and audibly engaging experience.