EduLightHub Mini Project - Step-by-Step Code Explanation

1. HTML Layout and Tailwind CSS

The layout is structured using Tailwind CSS utility classes:
- The <header> contains a fixed navigation bar using `fixed`, `bg-white`, `shadow-sm`, and `w-full`.
- `container mx-auto px-4` centers content and adds padding.
- Navigation links use `hover:text-indigo-600` to add hover effects.
- Dropdown menus and modals are initially hidden using `hidden`, then toggled via JavaScript.
- `font-sans`, `bg-gray-50` and similar classes are used for consistent typography and background.

2. JavaScript: User Dropdown Toggle

The dropdown menu for the user profile works as follows:
Step 1: JavaScript fetches elements using `getElementById` for userDropdown and dropdownMenu.
Step 2: When the user clicks the profile icon, it toggles the `hidden` class on the dropdown.
Step 3: Clicking outside the dropdown will hide it again using `document.addEventListener`.

3. JavaScript: Login/Signup Modal

Step 1: Clicking the login button triggers `openModal('loginModal')`, which removes the `hidden` class.
Step 2: `document.body.style.overflow = 'hidden'` disables page scrolling while modal is open.
Step 3: Clicking the close button or outside the modal calls `closeModal('loginModal')`, restoring scroll.

4. JavaScript: Course Filtering

Step 1: Two <select> elements allow filtering by category and level.
Step 2: `filterCourses()` loops through all `.course-card` elements.

Step 3: Cards are shown/hidden based on whether their `data-category` and `data-level` match the filters.

5. JavaScript: AI Chatbot Interaction

Step 1: On clicking the chatbot button, `chatbotModal` is displayed.
Step 2: The user enters a message and clicks 'Send' or presses Enter.
Step 3: The bot uses simple keyword-matching (e.g., 'course', 'assignment') to generate preset responses.
Step 4: All chat messages are dynamically added to the DOM.
Step 5: The modal can be closed via the X button or clicking outside.

6. JavaScript: Voice Input for Notes

Step 1: On clicking the microphone icon, SpeechRecognition API starts.
Step 2: The user's voice is transcribed and added to the notes textarea.
Step 3: If the browser doesn't support the API, an alert notifies the user.
Step 4: Voice feedback is shown via icon and color changes using Tailwind classes.