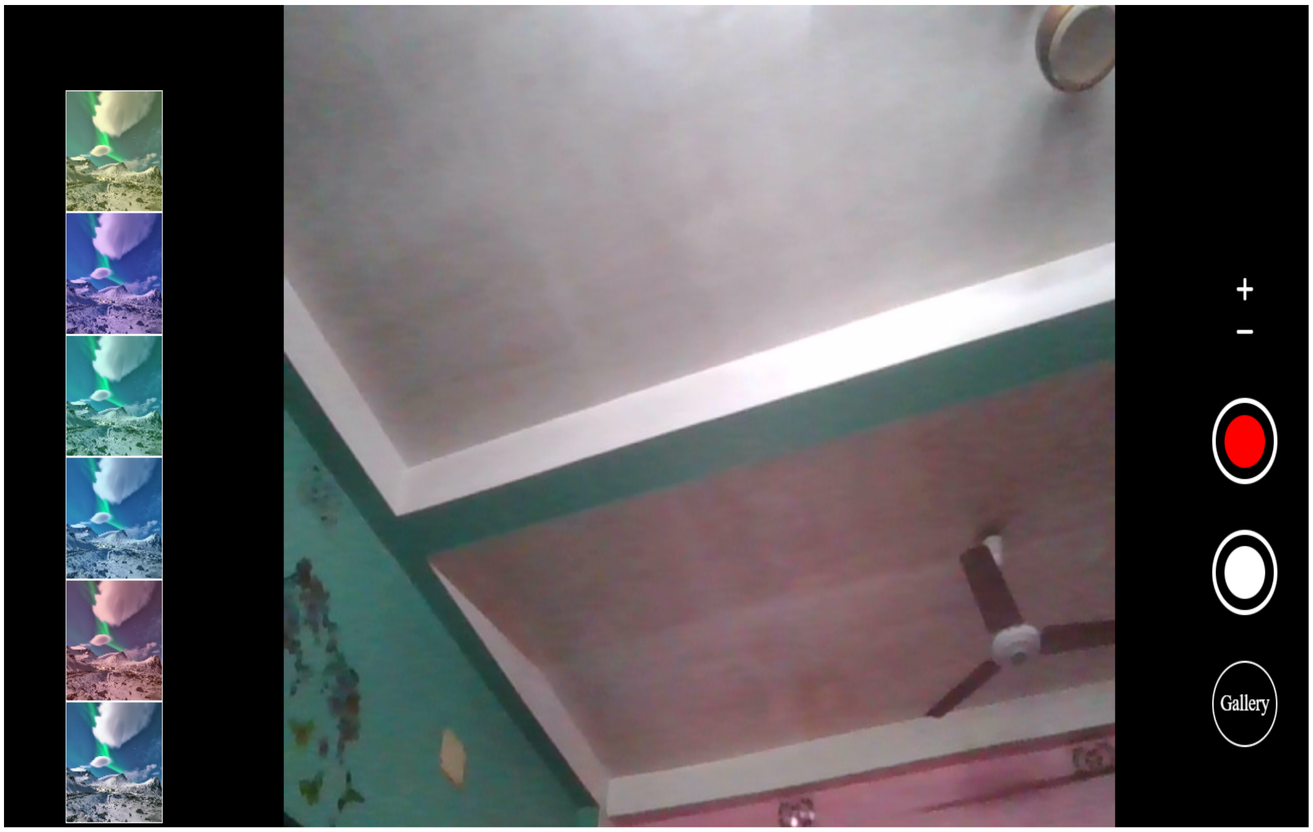

Video camera with gallery

Project Link : https://github.com/jaydip1235/placewit-js/tree/master/Video-Camera

Jaydip Dey
Jadavpur University
https://linktr.ee/jaydipdey

This is a video camera and gallery application that provides functionalities for interacting with the IndexedDB database for storing and retrieving media files. It extends the functionality of

- Video capturing
- Image capturing
- Zoom in and Zoom out
- Applying filters
- Downloading media
- Delete stored media

Code Summary

The application is written in JavaScript and interacts with the Document Object Model (DOM) to handle user interactions and perform operations. The code uses the built-in MediaDevices API to capture video and image data, along with basic elements of HTML and CSS to display the resulting media.The code works with the IndexedDB database to retrieve media files (images and videos) and present them on the webpage. It also provides options to download the displayed media or remove them from the database.

Detailed Breakdown of the file script.js

1. Variable Initialization: The script begins by initializing several variables that reference important DOM elements such as the video display element, record button, capture button, filters, zoom buttons, and the gallery button.
2. Event Listeners: Event listeners are added to the gallery button for redirecting to a gallery page, and to the zoom buttons for increasing or decreasing the video zoom level, capped within certain bounds.
3. Media Stream Capture: An asynchronous function is declared and immediately invoked to get access to the video stream from the user's camera using the navigator.mediaDevices.getUserMedia() function. Once the media stream is available, it is set as the source for the video display element.
4. MediaRecorder Setup: An instance of the MediaRecorder interface is created with the media stream. This object is used to record the video. Event listeners are added for the onstart, ondataavailable, and onstop events, enabling the logging of these events and the handling of video data when available.
5. Record and Capture Button Event Listeners: The record button toggles the recording state, and the capture button triggers the capturePhotoFun() function which captures a snapshot of the video feed and processes it.

6. Filter Handling: Each filter element is given a click event listener that creates a filter div and sets its background color to that of the selected filter.
7. Zoom Control: The click events for zoom in and zoom out buttons increase and decrease the current zoom level respectively, within certain bounds.
8. Photo Capturing: The capturePhotoFun() function captures a still image from the video feed, applies the current filter, and saves the image data to the IndexedDB database.
9. Media Recording: The recordMediaFun() function starts or stops the video recording based on the current recording state, and toggles the recording state.
10. Media Addition to DB: The addMedia() function adds media files to the IndexedDB database.

Detailed Breakdown of file gallery.js

1. Database Initialization: The script begins by initializing an IndexedDB database named "Gallery". A new object store "Media" is created with "mid" as the keyPath in the 'onupgradeneeded' event. In the 'onsuccess' event, the script calls the fetchMedia() function to retrieve all media from the database and display them. An alert is triggered in case of an error.
2. Fetch Media Function: This function opens a read only transaction on the "Media" object store and then opens a cursor to iterate over the stored media files. For each media file, based on the type of media (photo/video), it calls the respective function to append the media to the gallery.
3. Append Photo Function: The appendPhoto() function creates a div containing the media image and associated buttons for download and delete. It also assigns event listeners to these buttons for downloading and deleting the media file. The div is then appended to the gallery section of the page.
4. Append Video Function: The appendVideo() function creates a div containing the media video and associated buttons for download and delete. As with the appendPhoto function, event listeners are attached to these buttons for downloading and deleting the media file. The div is then appended to the gallery section of the page.
5. Download Media Function: The downloadMedia() function creates an 'a' element and uses it to trigger a download of the media file when clicked. The media type is determined to set the appropriate download filename and href attribute.

6. Delete Media Function: The deleteMedia() function removes a media file from the IndexedDB and also removes the associated div from the page. It achieves this by opening a readwrite transaction on the "Media" object store and calling the delete method with the media id (mid) as the argument.