Video Annotation Tool - Take Home Assignment

Project Description

Build a web-based video annotation tool that allows users to watch videos and add timestamped annotations. Think of it as a simplified version of YouTube's comment system, but with annotations tied to specific moments and positions in the video.

Requirements

Core Functionality

1. Video Player Interface

Create a custom video player with the following features:

Basic Controls:

- Play/pause button
- Progress bar with seek functionality
- Current time / total duration display
- Fullscreen toggle

Design Guidelines:

- Use YouTube's video player as inspiration for layout and styling
- Apply your own creative design choices for colors, spacing, and visual elements
- Ensure the interface is clean, modern, and user-friendly
- Make it responsive across desktop, tablet, and mobile devices

Enhanced Features:

- Frame-by-frame navigation (previous/next frame buttons)
- Playback speed control (0.5x, 1x, 1.25x, 1.5x, 2x)
- Keyboard shortcuts (spacebar for play/pause, arrow keys for seek)

2. Advanced Video Annotation System

Implement a sophisticated annotation system with drawing capabilities:

Drawing Tools:

- **Circle Tool:** Click and drag to draw circles of varying sizes
- **Rectangle Tool:** Click and drag to create rectangles/squares
- **Line Tool:** Click and drag to draw straight lines

- **Text Tool:** Click to add text annotations at specific positions

Annotation Interaction:

- **Selection:** Click on any annotation shape to select it (show selection handles)
- **Movement:** Drag selected annotations to reposition them on the video frame
- **Deletion:** Delete selected annotations using delete key or delete button
- **Undo/Redo:** Implement undo/redo functionality for all annotation actions

Timeline Integration:

- **Progress Bar Markers:** Show visual indicators on the progress bar where annotations exist
- **Timestamp Visibility:** When playing the video, annotations should appear only during their respective timestamps
- **Duration Control:** Each annotation should be visible for 2-3 seconds from its timestamp or until the next annotation

Visual Requirements:

- Semi-transparent overlays that don't obstruct video content
- Clear selection indicators (handles, borders) for selected annotations

Annotation Management:

- **Toolbar:** Provide a drawing toolbar with tool selection (circle, rectangle, line, text)
- **Properties Panel:** Show properties of selected annotation (color, timestamp, text content)
- **Annotation List:** Sidebar showing all annotations with thumbnails and timestamps
- **Video Pause Requirement:** Annotations can only be created when video is paused

Data Persistence:

- Save annotations to browser storage (localStorage)
- Annotations should persist between page refreshes

3. Backend API (Supporting Role)

Create a simple REST API to handle annotation data:

Required Endpoints:

```
GET    /api/annotations     - Retrieve all annotations for a video
POST   /api/annotations     - Create a new annotation
PUT    /api/annotations/:id - Update an existing annotation
DELETE /api/annotations/:id - Delete an annotation
```

Technical Requirements

Frontend Stack:

- **Framework:** React (preferred), Vue.js, or Angular
- **Styling:** Modern CSS, Sass/SCSS, or CSS-in-JS
- **State Management:** Redux, Context API
- **Build Tools:** Vite, or similar

Backend Stack:

- **Runtime:** Node.js with Express
- **Database:** MongoDB, or file-based storage

Video Source:

- Use any sample video of your choice (MP4 format, 30-60 seconds recommended)
- Host it locally or use a public URL
- Ensure it's appropriate for a professional assessment

Deliverables

1. Working Application

- **Frontend:** Deployed on Netlify, Vercel, or similar platform
- **Backend:** Deployed on Heroku, or similar platform

2. Source Code

- **Repository:** Clean, well-organized GitHub repository
- **Structure:** Separate frontend and backend code clearly

3. Documentation

Include a comprehensive README.md with:

- Project overview and features implemented
- Technology stack used
- API documentation with example requests/responses
- Screenshots demonstrating key features
- Any assumptions or design decisions made

4. Optional Enhancements (Bonus Points)

- **Testing:** Unit tests for annotation logic and drawing functions
- **Performance:** Canvas optimization for smooth drawing

- **Accessibility:** Keyboard shortcuts for drawing tools (C for circle, R for rectangle, etc.)
- **Export:** Ability to export annotations
- **Animation:** Smooth transitions for annotation appearance/disappearance
- **Innovation:** Implement a feature that you believe would significantly improve the video annotation experience, but isn't mentioned in the requirements above. This is your opportunity to showcase creativity and deep thinking about user needs.

Evaluation Criteria

Frontend Development (70% weight)

- **Code Quality:** Clean, readable, and well-structured code
- **Drawing Implementation:** Smooth and responsive annotation tools
- **User Experience:** Intuitive drawing interface and annotation management
- **Canvas/SVG Mastery:** Efficient rendering and interaction handling
- **State Management:** Complex state handling for undo/redo and annotation properties
- **Timeline Integration:** Seamless annotation visibility during video playback

Backend Development (30% weight)

- **API Design:** RESTful endpoints with proper HTTP methods
- **Data Handling:** Proper validation and error handling
- **Code Organization:** Clear structure and best practices
- **Integration:** Seamless connection with frontend

General Assessment

- **Attention to Detail:** Polish in both functionality and presentation
- **Problem Solving:** Creative solutions to technical challenges
- **Documentation:** Clear setup instructions and code comments
- **Best Practices:** Following modern development standards

Submission Guidelines

1. **Code Repository:** Share the GitHub repository link
2. **Live Demo:** Provide URL to the live project
3. **Documentation:** Ensure README.md covers all required sections
4. **Short Video:** Provide a link (Google Drive or similar) to a short 1-2 minute video demoing the application

Questions?

If you have any questions about the requirements or need clarification on any aspects of the assignment, reach out to **somadatta218@gmail.com**

Good luck, and we're excited to see what you build!