

Frontend Developer Task Sheet – Video Editing Platform (UI)

Project Title:

Build the Frontend for a Web-based Video Editing Platform

Objective:

We are building a browser-based video editing tool. Your task is to **design and develop the frontend** for a web interface that allows users to:

- Upload custom videos
- Rearrange audio and cutscenes
- Add subtitles, text overlays, and styled images
- Preview the edited video in real-time
- Export and download the final version

This project is focused entirely on the **frontend implementation**. You can use mock APIs or stubs where backend integration is not available.

Tech Stack (Strict Requirements)

You **must** use the following:

- **Next.js** (with app router preferred)
- **React.js**
- **Tailwind CSS** (use ShadCN UI for components like Dialogs, Buttons, Tabs, Toasts, etc.)

- **Redux Toolkit** (for state management, especially for video timeline, audio clips, subtitles, etc.)

Optional/Recommended:

- **React Player** / **HTML5 `<video>` tag**
 - **React Dropzone** (for drag-n-drop upload UI)
 - **React DnD** or similar lib for timeline scene rearrangement
-

UI Requirements

The UI should include the following **functional areas**:

1. Video Upload Section

- Drag-and-drop file upload zone
- Progress bar (simulated)
- Show thumbnail/preview of uploaded video

2. Video Timeline Interface

- Horizontal strip showing video frames or markers
- Add/remove "scenes" (mock trimming logic)
- Rearranging segments via drag-and-drop
- Mock cut/edit functions (non-functional but realistic UI)

3. Audio Management

- Show audio waveform (static/mock)
- Allow user to rearrange or mute specific audio segments

- Add background music (stub input)

4. Subtitles & Text Overlay

- Add subtitle input box
- Choose timing, font, position
- Support multiple subtitle blocks
- Overlay text options with font, size, color, and positioning

5. Image Overlay

- Upload and place a static image over the video
- Resizable and draggable image element
- Optional styles (border, opacity, animation)

6. Preview & Render Controls

- Display real-time preview using `<video>` tag or player
- Mock a "Render" button with loading animation
- Simulate download button for "exported" result

Tips

- Focus on the **experience** and **clean UX** rather than just functionality.
- Use mock APIs and simulate processing/rendering logic with loading indicators.
- Keep the code modular and readable — treat this like a mini production app.

Submission Requirements

To complete your submission, please provide the following:

1. **GitHub Repository Link**

Upload your project to a public or private GitHub repository. Make sure the repository:

- Contains all necessary source code and assets
- Includes a clear **README.md** with setup instructions and any relevant notes
- Has clear commit history that reflects your progress

2. **Google Drive Link with Demo Video**

Record a short screen recording (3–5 minutes) of your completed project demonstrating all key features. The video **must include a voice-over** explanation walking through:

- The main interface and functionality
- How each feature works (e.g., uploads, editing UI, previews)
- Any challenges or creative decisions made

3. Upload the video to Google Drive and share the link with public or restricted access (as preferred). Make sure sharing permissions are correct.

Deadline

The final submission is due **by May 1st at 6:00 PM IST**.

Important: Late submissions **will not be accepted** under any circumstances. Please ensure both the GitHub repo and the Google Drive demo link are shared **before the deadline**.