Here is the requirements for the problem:

You need to deliver small program (code can be on github or similar) that:

1. Input: A video file (e.g., MP4 format).
2. Output: Two separate audio tracks:
   * Track 1: Cleaned vocals (as much as possible).
   * Track 2: Background noises.
3. Implementation Guidelines:
   * Use any open-source libraries or tools you're familiar with.
   * While deep learning solutions like U-Net architectures are popular for such tasks, feel free to use any approach, including classical digital signal processing methods, if you believe they can be effective within the time constraints.
   * Remember, the primary focus is on separating vocals from background for STT improvement, so think about what kind of background noises might be most disruptive to STT systems and prioritize separating those.
4. Evaluation:
   * Briefly explain your approach and the rationale behind your choices.
   * If possible, demonstrate the effectiveness of your solution by running a simple STT on the original and cleaned audio tracks (you can use any open-source STT for this). Compare the transcription accuracy.
   * Discuss potential improvements and limitations of your solution.
5. Bonus (if time permits):
   * Discuss how this audio separation module might be integrated into a larger speech-to-text pipeline. Consider factors like real-time processing, scalability, and adaptability to different video sources.

Hints and Tips:

* Consider tools/libraries like TensorFlow, PyTorch, librosa, etc.
* There are pretrained models and solutions available for tasks like this. Familiarity with such solutions can be a starting point, but remember to cite sources if you borrow code or ideas.