

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8" />
<meta name="viewport"
content="width=device-width, initial-
scale=1" />
<title>PDF to Audiobook Converter</title>
<style>
  @import url('https://
fonts.googleapis.com/css2?
family=Inter&display=swap');
  body {
    font-family: 'Inter', sans-serif;
    margin: 0; padding: 0;
    background-color: #f0f2f5;
```

```
color: #222;
transition: background-color 0.3s, color
0.3s;
}
body.dark {
background-color: #121212;
color: #eee;
}
header {
display: flex;
justify-content: space-between;
align-items: center;
padding: 1rem 1.5rem;
background-color: #005aee;
color: white;
box-shadow: 0 2px 6px rgba(0, 90, 238,
0.5);
}
body.dark header {
background-color: #0041bb;
box-shadow: 0 2px 6px rgba(0, 65, 187,
```

```
0.5);  
}  
header h1 {  
    margin: 0; font-weight: 700; font-size:  
1.4rem;  
}  
#modeToggle {  
    background: none;  
    border: none;  
    color: white;  
    font-size: 1.4rem;  
    cursor: pointer;  
}  
main {  
    max-width: 720px;  
    margin: 1.5rem auto 3rem;  
    padding: 0 1rem;  
}  
label.upload-label {  
    display: inline-flex;  
    align-items: center;
```

```
gap: 0.5rem;
padding: 1rem 1.5rem;
border-radius: 12px;
background-color: white;
color: #005aee;
font-weight: 600;
font-size: 1.1rem;
cursor: pointer;
box-shadow: 0 6px 12px
rgba(0,90,238,0.15);
  transition: background-color 0.3s ease;
}
body.dark label.upload-label {
  background-color: #222;
  color: #66aaff;
  box-shadow: 0 6px 12px
rgba(0,90,238,0.6);
}
label.upload-label:hover {
  background-color: #e0eaff;
}
```

```
body.dark label.upload-label:hover {  
  background-color: #3355aa;  
}  
  
input[type="file"] {  
  display: none;  
}  
  
select, button {  
  margin-top: 1rem;  
  padding: 0.6rem 1rem;  
  font-size: 1rem;  
  border-radius: 8px;  
  border: 1px solid #ccc;  
  outline: none;  
  transition: all 0.3s ease;  
}  
  
body.dark select, body.dark button {  
  background-color: #222;  
  border-color: #555;  
  color: #eee;  
}  
  
select:focus, button:focus {
```

```
border-color: #005aee;
}
#textPreview {
  width: 100%;
  height: 180px;
  margin-top: 1rem;
  border-radius: 8px;
  padding: 0.8rem;
  font-family: monospace;
  font-size: 0.9rem;
  border: 1px solid #ccc;
  resize: vertical;
}
body.dark #textPreview {
  background-color: #222;
  border-color: #555;
  color: #eee;
}
#controls {
  margin-top: 1rem;
  display: flex;
```

```
    gap: 1rem;
}
#controls button {
    flex: 1;
    cursor: pointer;
    background-color: #005aee;
    color: white;
    border: none;
    box-shadow: 0 4px 10px
rgba(0,90,238,0.7);
}
body.dark #controls button {
    background-color: #0041bb;
    box-shadow: 0 4px 10px
rgba(0,65,187,0.7);
}
#controls button:disabled {
    background-color: #888;
    cursor: not-allowed;
    box-shadow: none;
}
```

```
#footer {
  text-align: center;
  margin-top: 3rem;
  padding: 1rem 0;
  color: #666;
  font-size: 0.85rem;
}
body.dark #footer {
  color: #999;
}
</style>
</head>
<body>
<header>
  <h1>PDF to Audiobook</h1>
  <button id="modeToggle" title="Toggle
Dark Mode" aria-label="Toggle Dark Mode">
    🌙 </button>
</header>
<main>
  <label for="pdfInput" class="upload-label">
```



```
tabindex="0">
```






Upload PDF

```
<input type="file" id="pdfInput"
accept="application/pdf" />
</label>
```

```
<div>
```

```
<label for="langSelect">Select
Language:</label>
<select id="langSelect" aria-label="Select
language">
  <option value="en-US">English</option>
  <option value="bn-BD">Bangla</option>
  <option value="ar-SA">Arabic</option>
</select>
</div>
```

```
<textarea id="textPreview"
placeholder="Extracted PDF text will
appear here..." readonly></textarea>
```

```
<div id="controls">
  <button id="playBtn" disabled aria-
label="Play audiobook"> Play</button>
  <button id="pauseBtn" disabled aria-
label="Pause audiobook"> Pause</
button>
  <button id="stopBtn" disabled aria-
label="Stop audiobook"> Stop</button>
</div>
</main>
<footer id="footer">© 2025 PDF to
Audiobook | Developed by ChatGPT</
footer>

<script src="https://cdn.jsdelivr.net/npm/
pdfjs-dist@3.8.162/build/pdf.min.js"></
script>
<script>
  const pdfjsLib = window['pdfjs-dist/build/
pdf'];
  pdfjsLib.GlobalWorkerOptions.workerSrc
```

```
= 'https://cdn.jsdelivr.net/npm/pdfjs-dist@3.8.162/build/pdf.worker.min.js';
```

```
const pdfInput =  
document.getElementById('pdfInput');  
const langSelect =  
document.getElementById('langSelect');  
const textPreview =  
document.getElementById('textPreview');  
const playBtn =  
document.getElementById('playBtn');  
const pauseBtn =  
document.getElementById('pauseBtn');  
const stopBtn =  
document.getElementById('stopBtn');  
const modeToggle =  
document.getElementById('modeToggle');  
const body = document.body;
```

```
let utterance = null;  
let isPlaying = false;
```

```
// Dark mode toggle
modeToggle.addEventListener('click', ()
=> {
  body.classList.toggle('dark');
  modeToggle.textContent =
body.classList.contains('dark') ? '☀️' : '🌙';
});
```

```
// Extract text from PDF
pdfInput.addEventListener('change',
async (e) => {
  const file = e.target.files[0];
  if (!file) return;
```

```
  textPreview.value = "Loading PDF and
extracting text... Please wait.";
  playBtn.disabled = true;
  pauseBtn.disabled = true;
  stopBtn.disabled = true;
```

```
    const arrayBuffer = await
file.arrayBuffer();
    const pdf = await
pdfjsLib.getDocument(arrayBuffer).promis
e;
    let fullText = "";

    for(let i=1; i<=pdf.numPages; i++){
        const page = await pdf.getPage(i);
        const content = await
page.getTextContent();
        const strings = content.items.map(item
=> item.str);
        fullText += strings.join(' ') + '\n\n';
    }

    textPreview.value = fullText.trim() || "No
extractable text found in this PDF.";
    if(fullText.trim()) playBtn.disabled =
false;
    });
```

```
// Speech functions
```

```
function resetButtons() {  
    playBtn.disabled = false;  
    pauseBtn.disabled = true;  
    stopBtn.disabled = true;  
}
```

```
function playSpeech() {  
    if(!textPreview.value) return;  
    if(window.speechSynthesis.speaking) {  
        window.speechSynthesis.cancel();  
        resetButtons();  
        return;  
    }
```

```
    utterance = new  
SpeechSynthesisUtterance(textPreview.val  
ue);  
    utterance.lang = langSelect.value;  
    utterance.rate = 1;  
    utterance.pitch = 1;
```

```
utterance.onstart = () => {
    isPlaying = true;
    playBtn.disabled = true;
    pauseBtn.disabled = false;
    stopBtn.disabled = false;
};
utterance.onend = () => {
    isPlaying = false;
    resetButtons();
};
utterance.onerror = () => {
    isPlaying = false;
    resetButtons();
    alert("Speech synthesis error
occurred.");
};

window.speechSynthesis.speak(utterance);
}
```

```
function pauseSpeech() {  
    if(window.speechSynthesis.spoken &&  
!window.speechSynthesis.paused) {  
        window.speechSynthesis.pause();  
        pauseBtn.disabled = true;  
        playBtn.disabled = false;  
    }  
}
```

```
function resumeSpeech() {  
    if(window.speechSynthesis.paused) {  
        window.speechSynthesis.resume();  
        pauseBtn.disabled = false;  
        playBtn.disabled = true;  
    }  
}
```

```
function stopSpeech() {  
    if(window.speechSynthesis.spoken) {  
        window.speechSynthesis.cancel();  
    }  
}
```



```
    resetButtons();  
  }  
}
```

```
playBtn.addEventListener('click', () => {  
  if(window.speechSynthesis.paused) {  
    resumeSpeech();  
  } else {  
    playSpeech();  
  }  
});
```

```
pauseBtn.addEventListener('click',  
pauseSpeech);  
stopBtn.addEventListener('click',  
stopSpeech);
```

```
</script>
```

```
</body>
```

```
</html>
```