**Immersive Story Reading Experience with Ambient Sound Integration**

**Team Members: 3rd October, 2024**

Pavan Kumar S – 21BCE1179

Shravan Venkatraman – 21BCE1200

Aravintakshan S A – 21BCE1137

### Abstract

The "Immersive Story Reading Experience with Ambient Sound Integration" project explores innovative methods to enhance user engagement and immersion in narrative experiences through Human-Computer Interaction (HCI). By integrating ambient sound effects into story narration, this project transforms traditional reading into a multi-sensory journey, allowing users to connect more deeply with the narrative. The application employs a sophisticated text-to-speech (TTS) engine that provides diverse voice profiles, catering to user preferences and enriching the emotional depth of the storytelling.

Central to the project's design is context-aware sound integration, which intelligently identifies keywords or phrases within the narrative to trigger specific ambient sounds—such as rain, thunder, forest ambiance, and ocean waves—thus ensuring that the auditory environment aligns with the story's themes. This thoughtful sound integration enhances the narrative experience while maintaining a focus on user engagement.

Dynamic sound management further enriches the experience by balancing ambient sound volume with TTS narration, utilizing fade-in and fade-out effects to create seamless auditory transitions. This attention to detail aligns with user-centered design principles, ensuring that the user experience is both cohesive and immersive. User customization is another crucial aspect of the application, empowering users to tailor their auditory experience by selecting sound types and adjusting volumes. This flexibility promotes personalization, enhancing user satisfaction and engagement with the storytelling process.

The technical implementation includes a robust stack involving Vanilla JavaScript, Django, the Google Text-to-Speech API, and advanced Natural Language Processing (NLP) techniques. By leveraging these technologies, the project aims to redefine the reading experience, emphasizing multi-sensory engagement and customization, which are vital in contemporary HCI practices. Ultimately, this innovative application aspires to create a captivating platform that reimagines storytelling for modern audiences.

**Code Reference:**

**Login.html**  
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Login</title>

<link rel="stylesheet" href="css/login-styles.css">

</head>

<body>

<div class="box">

<form action="">

<div class="input-box">

<h2>Log In</h2>

<input type="text" required>

<span>Username</span>

<i></i>

</div>

<div class="input-box">

<input type="password" required>

<span>Enter Password</span>

<i></i>

</div>

<input type="submit" value="Login">

<div class="links">

<a href="#">Help</a>

<a href="signup.html">Sign Up</a>

</div>

</form>

</div>

</body>

</html>

**Signup.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Signup</title>

<link rel="stylesheet" href="css/login-styles.css">

</head>

<body>

<div class="box">

<form action="">

<div class="input-box">

<h2>Sign Up</h2>

<input type="text" required>

<span>Name</span>

<i></i>

</div>

<div class="input-box">

<input type="username" required>

<span>Username</span>

<i></i>

</div>

<!-- <div class="input-box">

<input type="password" required>

<span>Enter Password</span>

<i></i>

</div>

<div class="input-box"></div>

<input type="confirm\_password" required>

<span>Confirm Password</span>

<i></i>

</div> -->

<input type="submit" value="Login">

<div class="links">

<a href="#">Help</a>

<a href="login.html">Log in</a>

</div>

</form>

</div>

</body>

</html>

**Dashboard.html**  
  
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Frontend Book Application</title>

<link rel="stylesheet" href="./css/dashboard-styles.css">

</head>

<body>

<div class="container">

<!-- Navbar -->

<nav class="navbar">

<div class="logo">ISRASI</div>

<div class="search-bar">

<input type="text" placeholder="Search books online">

<button>🔍</button>

</div>

<div class="icons">

<button class="settings-icon">⚙️</button>

<button class="profile-icon">👤</button>

</div>

</nav>

<!-- Library Section -->

<section class="library">

<h1>MY LIBRARY</h1>

<div class="book-grid">

<!-- Book 1 -->

<div class="book-card" onclick="openBook('book1')">

<div class="book-cover"></div>

<p class="book-title">Book 1</p>

<p id="book1-progress" class="book-progress">progress 0%</p>

</div>

<!-- Book 2 -->

<div class="book-card" onclick="openBook('book2')">

<div class="book-cover"></div>

<p class="book-title">Book 2</p>

<p id="book2-progress" class="book-progress">progress 0%</p>

</div>

</div>

</section>

<!-- Book Content Modal -->

<div id="bookModal" class="modal">

<div class="modal-content">

<span class="close" onclick="closeModal()">&times;</span>

<h2 id="book-title"></h2>

<h3 id="chapter-title"></h3>

<p id="page-content"></p>

<div class="controls">

<button onclick="previousPage()">Previous</button>

<button onclick="nextPage()">Next</button>

</div>

<button class="speaker-button" onclick="readChapter()">🔊 Read Aloud</button>

<button class="back-button" onclick="closeModal()">Back to Library</button>

</div>

</div>

</div>

<script src="./scripts/dashboard-script.js"></script>

</body>

</html>

**Stylesheets:**

**Login-stlyes.css**

@import url('https://fonts.googleapis.com/css2?family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&display=swap');

\*{

margin: 0;

padding: 0;

box-sizing: border-box;

font-family: 'Poppins', sans-serif;

}

body{

width: 100%;

display: flex;

justify-content: center;

align-items: center;

min-height: 100vh;

background: url('https://i.postimg.cc/XYjWrv36/dark-hexagonal-background-with-gradient-color\_79603-1409.jpg') no-repeat;

background-size: cover;

background-position: center;

}

.box{

position: relative;

width: 370px;

height: 450px;

background: #1c1c1c;

border-radius: 50px 5px;

overflow: hidden;

}

.box::before{

content: '';

position: absolute;

top: -50%;

left: -50%;

width: 370px;

height: 450px;

background: linear-gradient(60deg, transparent, #45f3ff, #45f3ff);

transform-origin: bottom right;

animation: animate 6s linear infinite;

}

.box::after{

content: '';

position: absolute;

top: -50%;

left: -50%;

width: 370px;

height: 450px;

background: linear-gradient(60deg, transparent, #d9138a, #d9138a);

transform-origin: bottom right;

animation: animate 6s linear infinite;

animation-delay: -3s;

}

@keyframes animate{

0% { transform: rotate(0deg);}

100% { transform: rotate(360deg);}

}

form{

position: absolute;

inset: 2px;

border-radius: 50px 5px;

background: #28292d;

z-index: 10;

padding: 30px 30px;

display: flex;

flex-direction: column;

}

h2{

color: #45f3ff;

font-size: 35px;

font-weight: 500;

text-align: center;

}

.input-box{

position: relative;

width: 300px;

margin-top: 35px;

}

.input-box input{

position: relative;

width: 100%;

padding: 20px 10px 10px;

background: transparent;

border: none;

outline: none;

color: #23242a;

font-size: 1em;

letter-spacing: .05em;

z-index: 10;

}

input[type="submit"]{

font-size: 20px;

border: none;

outline: none;

background: #45f3ff;

padding: 5px;

margin-top: 40px;

border-radius: 90px;

font-weight: 600;

cursor: pointer;

}

input[type="submit"]:active{

background: linear-gradient(90deg, #45f3ff, #d9138a);

opacity: .8;

}

.input-box span{

position: absolute;

left: 0;

padding: 20px 10px 10px;

font-size: 1em;

color: #8f8f8f;

pointer-events: none;

letter-spacing: .05em;

transition: .5s;

}

.input-box input:valid ~ span,

.input-box input:focus ~ span{

color: #45f3ff;

transform: translateX(-10px) translateY(-30px);

font-size: .75em;

}

.input-box i{

position: absolute;

left: 0;

bottom: 0;

width: 100%;

height: 2px;

background: rgb(69, 243, 255);

border-radius: 4px;

transition: .5s;

pointer-events: none;

z-index: 9;

}

.input-box input:valid ~ i,

.input-box input:focus ~ i{

height: 44px;

background: rgba(69, 243, 255, .5);

}

.links{

display: flex;

justify-content: space-between;

}

.links a{

margin: 25px 0;

font-size: 1em;

color: #8f8f8f;

text-decoration: none;

}

.links a:hover,

.links a:nth-child(2){

color: #45f3ff;

}

.links a:nth-child(2):hover{

color: #d9138a;

}

**Dashboard-styles:**

.container {

font-family: Arial, sans-serif;

}

.navbar {

display: flex;

justify-content: space-between;

align-items: center;

padding: 10px 20px;

background-color: #333;

color: white;

}

.search-bar input {

padding: 5px;

}

.icons button {

margin-left: 10px;

}

.library {

padding: 20px;

}

.book-grid {

display: flex;

justify-content: space-around;

padding-top: 20px;

}

.book-card {

width: 150px;

height: 200px;

background-color: #333;

color: white;

display: flex;

flex-direction: column;

justify-content: center;

align-items: center;

cursor: pointer;

}

.book-title {

margin-top: 10px;

font-size: 18px;

}

.book-progress {

margin-top: 5px;

font-size: 14px;

}

/\* Modal styles \*/

.modal {

display: none;

position: fixed;

z-index: 1;

left: 0;

top: 0;

width: 100%;

height: 100%;

background-color: rgba(0,0,0,0.5);

justify-content: center;

align-items: center;

}

.modal-content {

background-color: #fff;

padding: 20px;

border-radius: 5px;

width: 400px;

text-align: center;

}

.close {

position: absolute;

right: 10px;

top: 10px;

cursor: pointer;

}

.controls button {

margin: 10px;

}

.back-button {

margin-top: 20px;

padding: 10px 20px;

background-color: #333;

color: white;

border: none;

cursor: pointer;

font-size: 16px;

border-radius: 5px;

}

.back-button:hover {

background-color: #555;

}

.speaker-button {

margin-top: 10px;

padding: 10px;

background-color: #007BFF;

color: white;

border: none;

cursor: pointer;

font-size: 18px;

border-radius: 5px;

}

.speaker-button:hover {

background-color: #0056b3;

}

**Scripts:**

**Dashboard-script.js**

const books = {

book1: {

title: "Book 1",

chapters: [

{ title: "Chapter 1", page: "This is the content of Chapter 1 of Book 1." },

{ title: "Chapter 2", page: "This is the content of Chapter 2 of Book 1." }

]

},

book2: {

title: "Book 2",

chapters: [

{ title: "Chapter 1", page: "This is the content of Chapter 1 of Book 2." },

{ title: "Chapter 2", page: "This is the content of Chapter 2 of Book 2." }

]

}

};

let currentBook = null;

let currentChapterIndex = 0;

function openBook(bookId) {

currentBook = books[bookId];

currentChapterIndex = 0;

showChapter();

document.getElementById('bookModal').style.display = 'flex';

loadProgress(bookId);

}

function showChapter() {

const chapter = currentBook.chapters[currentChapterIndex];

document.getElementById('book-title').innerText = currentBook.title;

document.getElementById('chapter-title').innerText = chapter.title;

document.getElementById('page-content').innerText = chapter.page;

const prevButton = document.querySelector('.controls button:first-of-type');

prevButton.style.display = currentChapterIndex === 0 ? 'none' : 'inline-block';

const nextButton = document.querySelector('.controls button:last-of-type');

nextButton.style.display = currentChapterIndex === currentBook.chapters.length - 1 ? 'none' : 'inline-block';

}

/\*

let speechSynthesisUtterance = null;

function readChapter() {

const chapterText = "Hello! This is a test for text-to-speech functionality."; // Hardcoded for testing

// Check if the browser supports speech synthesis

if (!window.speechSynthesis) {

console.error("Speech Synthesis API not supported by this browser.");

alert("Your browser does not support text-to-speech functionality.");

return;

}

// Cancel any existing speech synthesis (if a user clicks multiple times)

if (speechSynthesisUtterance) {

window.speechSynthesis.cancel();

}

// Create a new SpeechSynthesisUtterance

speechSynthesisUtterance = new SpeechSynthesisUtterance(chapterText);

speechSynthesisUtterance.lang = 'en-US'; // Set language

speechSynthesisUtterance.volume = 1; // Volume (0 to 1)

speechSynthesisUtterance.pitch = 1; // Pitch (0 to 2)

speechSynthesisUtterance.rate = 1; // Rate (0.1 to 10)

// Attempt to set the voice

const voices = window.speechSynthesis.getVoices();

if (voices.length > 0) {

speechSynthesisUtterance.voice = voices[0]; // Use the first available voice

} else {

console.warn("No voices available. Using default system voice.");

}

console.log("Starting to speak...");

// Start speaking the text

window.speechSynthesis.speak(speechSynthesisUtterance);

// Log when speech ends

speechSynthesisUtterance.onend = () => {

console.log("Speech has finished.");

};

// Log any errors

speechSynthesisUtterance.onerror = (event) => {

console.error("Speech synthesis error:", event);

};

}

// Load voices when the page is loaded

window.speechSynthesis.onvoiceschanged = () => {

console.log("Voices loaded.");

};

\*/

function closeModal() {

document.getElementById('bookModal').style.display = 'none';

// Stop any ongoing speech when the modal is closed

if (speechSynthesisUtterance) {

window.speechSynthesis.cancel();

}

}

function previousPage() {

if (currentChapterIndex > 0) {

currentChapterIndex--;

showChapter();

saveProgress();

}

}

function nextPage() {

if (currentChapterIndex < currentBook.chapters.length - 1) {

currentChapterIndex++;

showChapter();

saveProgress();

}

}

function saveProgress() {

const progress = ((currentChapterIndex + 1) / currentBook.chapters.length) \* 100;

localStorage.setItem(currentBook.title + '\_progress', progress);

document.getElementById(currentBook.title.toLowerCase().replace(' ', '') + '-progress').innerText = `progress ${progress.toFixed(0)}%`;

}

function loadProgress(bookId) {

const storedProgress = localStorage.getItem(books[bookId].title + '\_progress');

if (storedProgress) {

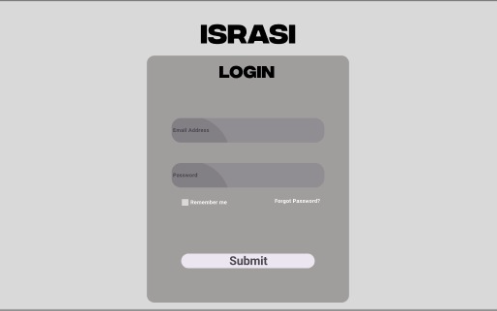
document.getElementById(bookId + '-progress').innerText = `progress ${parseFloat(storedProgress).toFixed(0)}%`;

}

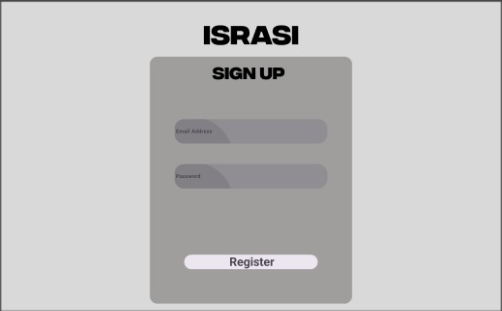
}

UI:

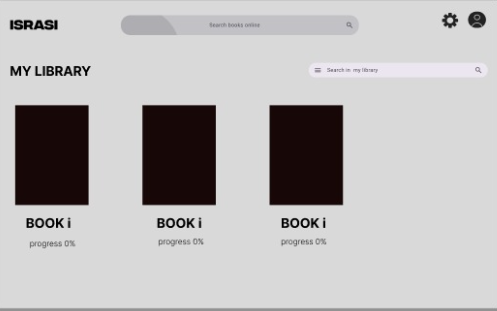
**Login page:**



**Signup page:**



**Dashboard:**



**Reader:**

