

# Generative AI Historical Scene Explorer

MetaChronicle is a learning tool powered by generative AI, allowing users to explore events through varying perspectives and emotional lenses. Drawing on research into how perspective and emotion shape narratives, the project encourages critical thinking and deeper engagement.

## Tools:

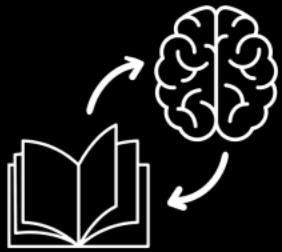


## Keywords:

#Generative AI #Interactive Storytelling #Emotional Engagement  
#Critical Thinking #UI/UX Design

## Website:

<https://canacechen.com/metachronicle.html>



## Interactive Storytelling and Emotional Engagement

Designing MetaChronicle's narrative approach involved blending interactive storytelling techniques that incorporate emotional resonance and shifting perspectives. Research into narrative theory and digital storytelling emphasises the importance of engaging users on both cognitive and affective levels (Ryan, 2006). By allowing users to experience historical events through multiple roles and emotional filters, the platform aims to enhance empathy and critical thinking. From an educational point of view, this approach also improves comprehension and retention, as "we feel, therefore we learn" (Immordino-Yang & Damasio, 2007, p. 5). Applying these insights, the design decisions included scene-by-scene narrative progression with viewpoint and emotional variability, thereby challenging preconceived notions and fostering a more nuanced, comprehensive understanding of the scenes (Wineburg, 2001).

## AI-Driven Narrative Possibilities

Integrating AI-driven language and image generation models opened up new avenues for dynamic, customised storytelling. It enables narrative experiences that extend beyond pre-scripted sources of text and imagery, allowing for virtually infinite possibilities in constructing historical scenes and perspectives. Large-scale generative models can rapidly generate coherent and contextually appropriate responses, including both narrative elements and visuals, thereby streamlining the creative process (Radford et al., 2019; Ramesh et al., 2021). By leveraging these capabilities, the platform can offer personalised experiences for each user's selected event, perspective, and emotional filter, thus moving beyond static historical representations to a more fluid, responsive storytelling approach, while minimising production time and costs associated with traditional content creation methods (Katsamakas and Sanchez-Cartas, 2024).



## Ethical and Societal Considerations in AI

Developing MetaChronicle required an awareness of the ethical and societal implications of employing generative AI for historical storytelling. Biases embedded in training data and the risk of misrepresenting marginalised voices demanded careful consideration (Crawford & Paglen, 2019). Additionally, AI-generated narratives might reproduce or challenge systemic inequities (Benjamin, 2019). These insights shaped the decision to incorporate multiple perspectives, prompt users to question their assumptions, and implement accessibility features. It also guided the content moderation strategies to mitigate misinformation and ensure that the platform promotes a respectful, inclusive representation of historical events.

## Claim of AI Usage

This project used ChatGPT-4o API to generate perspective and emotion selections and suggestions, narratives, and prompts for image generation, and Stable Diffusion 3.5 to generate images for each scene and the website background image.

“

# Platform Design

- An interactive tool that allows users to explore historical events through multiple perspectives and emotional filters.
- Users can experience stories from different viewpoints (e.g., soldier, civilian, leader) and emotional tones (e.g., fear, hope, anger).
- Combines AI-generated narratives and visuals, enabling deeper engagement and understanding of context and emotion.

## Primary users

Students, educators, history enthusiasts, researchers, and museum-goers.

## Design for Different User Needs

### Age Groups

- Younger users: Simplified narratives, engaging exploration of historical events.
- Adults/researchers: Detailed accounts, nuanced emotional filters, and deeper contextual information.

### Level of Interests

- Casual users: Brief, engaging overviews with visuals and emotions.
- Enthusiasts/researchers: Interactive deep dives with customisable content

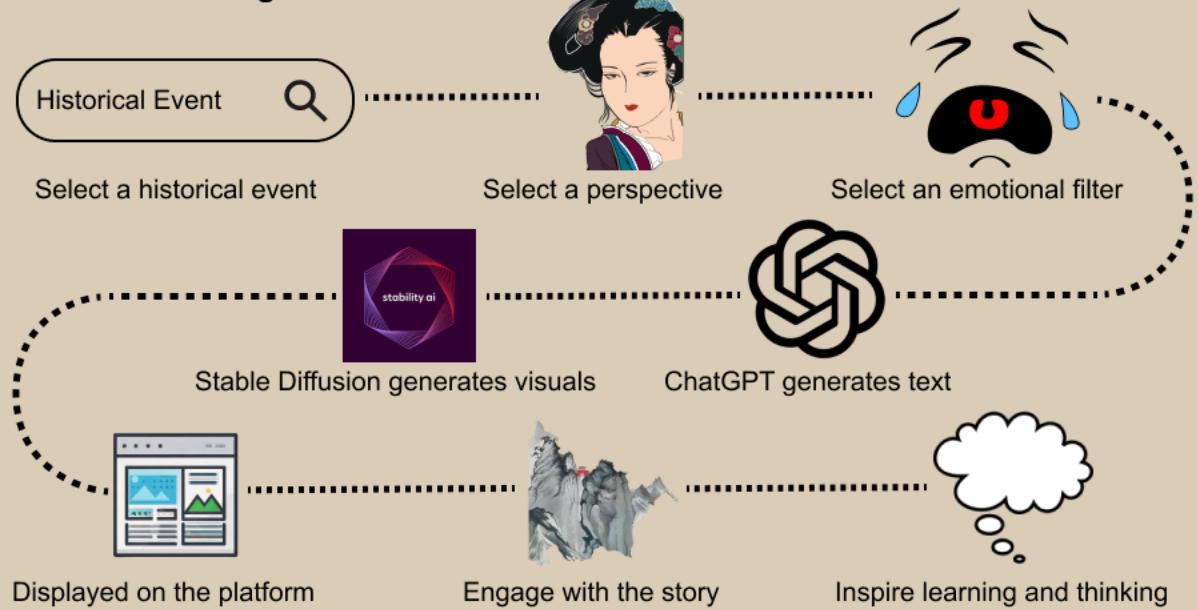
### Accessibility Features

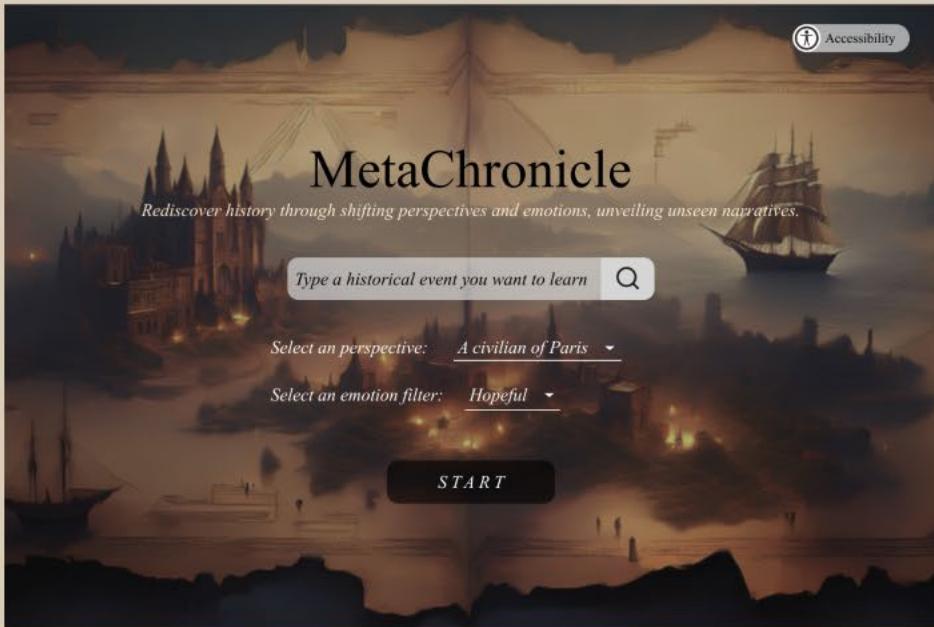
- Visual/audio descriptions for visually or hearing-impaired users.
- Simple controls for users with limited technical skills or disabilities.

## Technical components

Frontend Interface (web platform/UI): HTML, CSS, JavaScript  
Backend Server (application logic): Server-Side Framework (Express.js), Logic Layer  
Generative AI Integration: Language Model API (ChatGPT), Text-to-Image Model API (e.g., Stable Diffusion)

## Interaction Design





## UX + UI Design

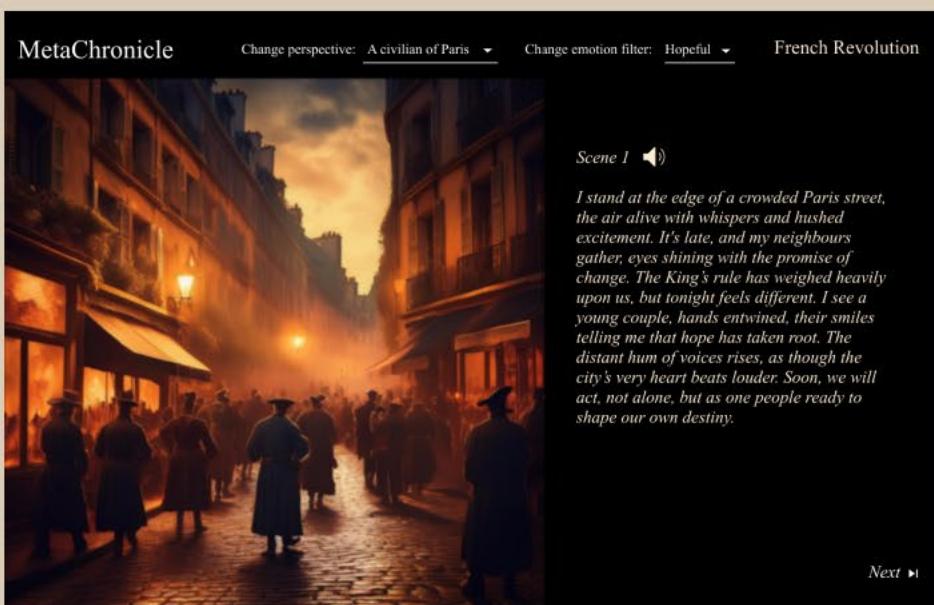
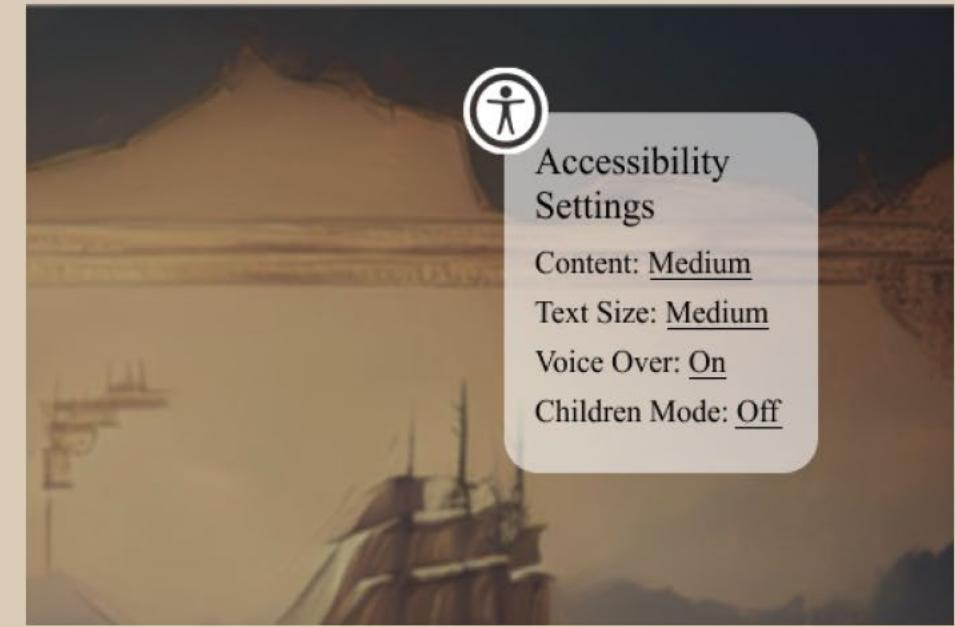
### 1. Select Your Experience

Choose a historical event, pick a perspective, and set an emotional filter. Press "Start" to begin the journey.

### 2. Adjust Accessibility

Before starting, refine the experience by changing content complexity, text size, or enabling voice-over or children's mode.

These settings ensure the experience suits personal needs.

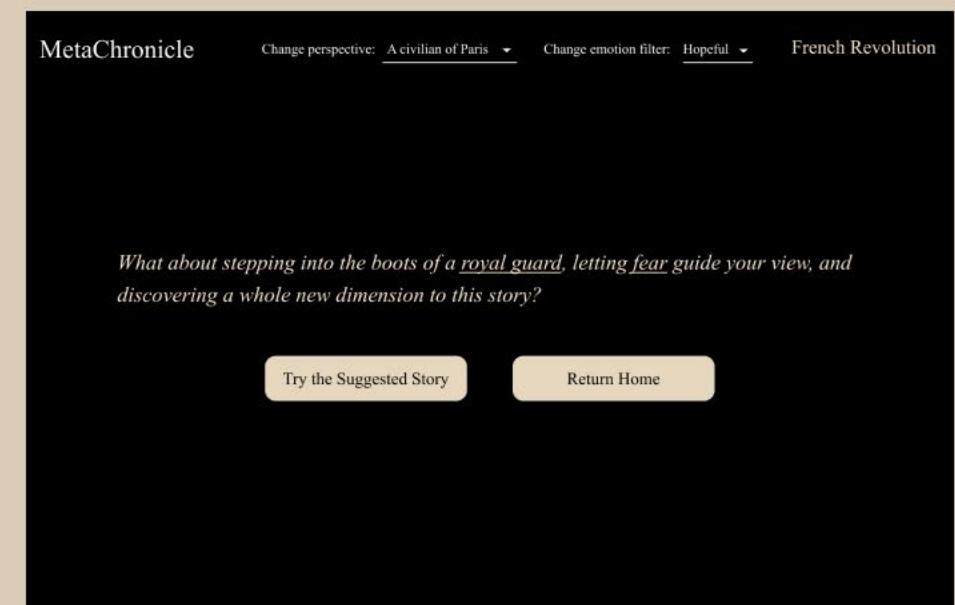


### 3. Immerse in the Story

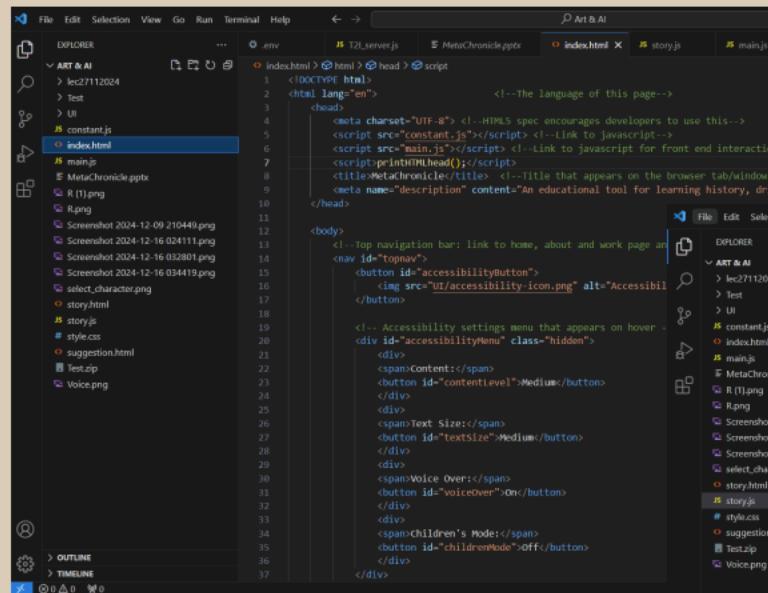
As the story unfolds, see dynamically generated images with the narratives side by side. Voice-over, perspective, and emotional filters remain adjustable. Press "Next" to continue the story.

### 4. Explore Alternatives

At the end, a different perspective and emotional filter combination is proposed to challenge initial assumptions. User can choose to explore further, or return home.



# Implementation

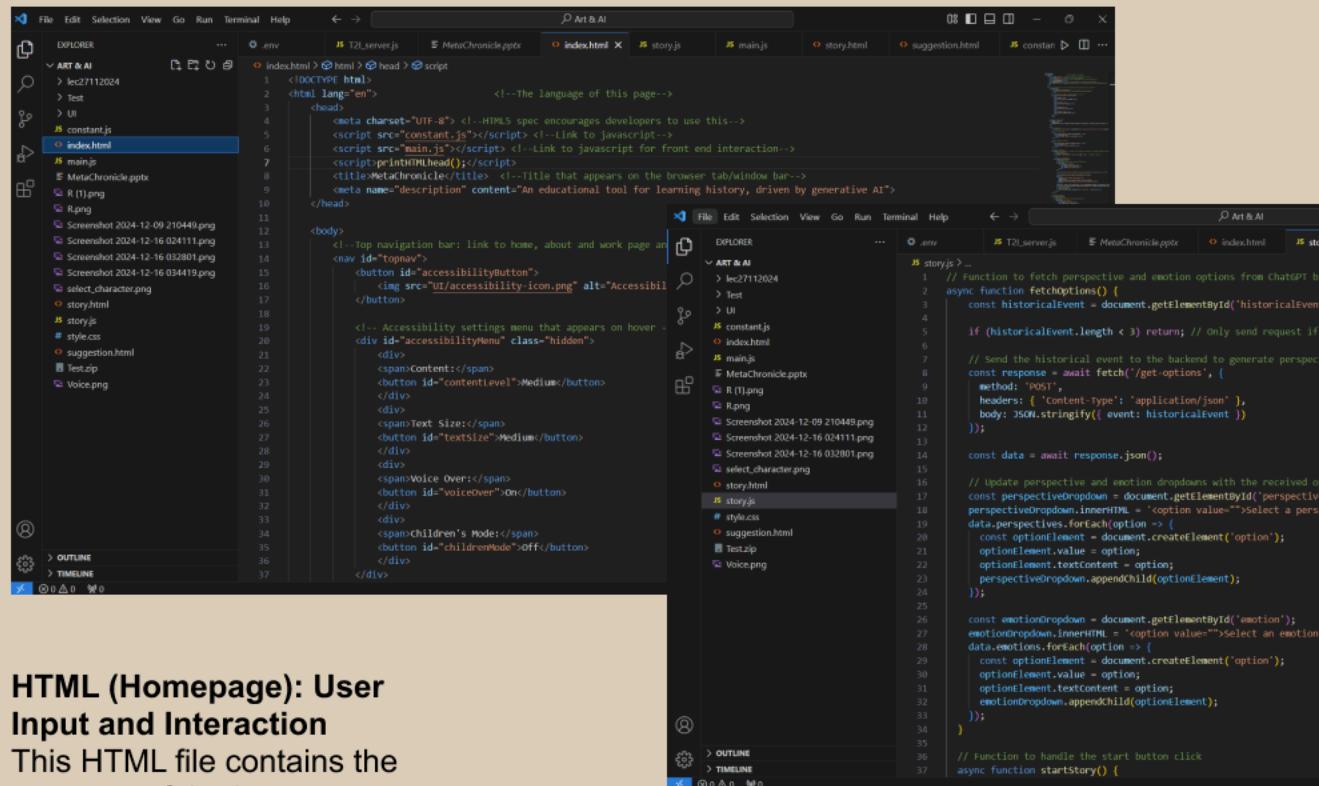


```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8"> <!-- HTML spec encourages developers to use this-->
    <script src="constant.js"></script> <!-- Link to javascript for front end interaction-->
    <script src="main.js"></script> <!-- Link to javascript for front end interaction-->
    <script>printHTMLHead();</script>
    <title>MetaChronicle</title> <!-- Title that appears on the browser tab/window bar-->
    <meta name="description" content="An educational tool for learning history, driven by generative AI">
  </head>
  <body>
    <!-- Top navigation bar: link to home, about and work page -->
    <nav id="topnav">
      <button id="accessibilitybutton">
        
      </button>
      <!-- Accessibility settings menu that appears on hover -->
      <div id="accessibilityMenu" class="hidden">
        <div>
          <span>Content:</span>
          <button id="contentlevel">Medium</button>
        </div>
        <div>
          <span>Text Size:</span>
          <button id="textsize">Medium</button>
        </div>
        <div>
          <span>Voice Over:</span>
          <button id="voiceover">On</button>
        </div>
        <div>
          <span>Children's Mode:</span>
          <button id="childrenMode">Off</button>
        </div>
      </div>
    </nav>
  </body>

```

## HTML (Homepage): User Input and Interaction

This HTML file contains the structure of the homepage, where users input a historical event, select a perspective, and choose an emotional filter. It includes dropdown menus for dynamic options and a button to start the story. The layout is designed to be simple and intuitive for user interaction.



```
async function fetchOptions() {
  const historicalEvent = document.getElementById('historicalEventInput').value;

  if (historicalEvent.length < 3) return; // Only send request if the input is sufficiently long

  // Send the historical event to the backend to generate perspective and emotion options
  const response = await fetch('/get-options', {
    method: 'POST',
    headers: { 'content-type': 'application/json' },
    body: JSON.stringify({ event: historicalEvent })
  });

  const data = await response.json();

  // Update perspective and emotion dropdowns with the received options
  const perspectiveDropdown = document.getElementById('perspective');
  perspectiveDropdown.innerHTML = <option value="">Select a perspective</option>;
  data.perspectives.forEach(option => {
    const optionElement = document.createElement('option');
    optionElement.value = option;
    optionElement.textContent = option;
    perspectiveDropdown.appendChild(optionElement);
  });

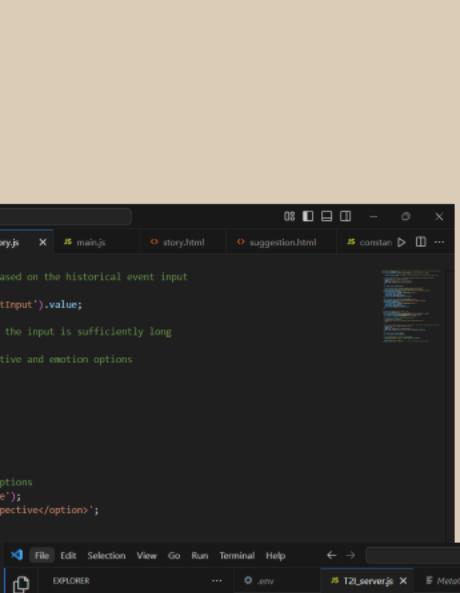
  const emotionDropdown = document.getElementById('emotion');
  emotionDropdown.innerHTML = <option value="">Select an emotion</option>;
  data.emotions.forEach(option => {
    const optionElement = document.createElement('option');
    optionElement.value = option;
    optionElement.textContent = option;
    emotionDropdown.appendChild(optionElement);
  });
}

// Function to handle the start button click
async function startStory() {

```

## JavaScript: Frontend Interaction Logic

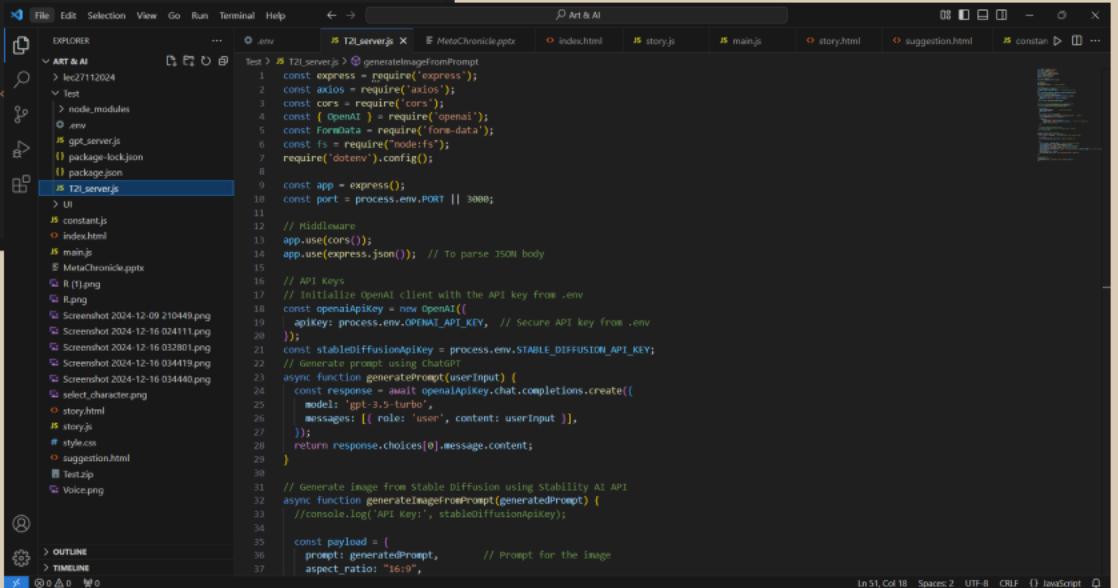
This JavaScript file handles the frontend logic for fetching perspective and emotional filter options based on user input. It sends the historical event to the backend and updates the dropdowns dynamically. Upon clicking "Start Story", it sends the selections to generate the first scene and navigates the user to the story page.



```
async function startStory() {
  const response = await fetch('/generateImageFromPrompt', {
    method: 'POST',
    headers: { 'content-type': 'application/json' },
    body: JSON.stringify({ perspective: 'present', emotion: 'neutral' })
  });

  const data = await response.json();

  // Update the story page with the generated image
  window.location.href = 'story.html';
}
```



```
const express = require('express');
const axios = require('axios');
const cors = require('cors');
const { OpenAI } = require('openai');
const FormData = require('form-data');
const fs = require('node:fs');
const dotenv = require('dotenv').config();

const app = express();
const port = process.env.PORT || 3000;

// Middleware
app.use(cors());
app.use(express.json()); // To parse JSON body

// API Keys
// Initialize OpenAI client with the API key from .env
const openaiApikey = new OpenAI({
  apiKey: process.env.OPENAI_API_KEY, // Secure API key from .env
});
const stableDiffusionApiKey = process.env.STABLE_DIFFUSION_API_KEY;
// Generate prompt using ChatGPT
async function generatePrompt(userInput) {
  const response = await openaiApikey.chat.completions.create({
    model: 'gpt-3.5-turbo',
    messages: [ { role: 'user', content: userInput } ],
  });
  return response.choices[0].message.content;
}

// Generate image from Stable Diffusion using Stability AI API
async function generateImageFromPrompt(generatedPrompt) {
  //console.log('API Key:', stableDiffusionApiKey);

  const payload = {
    prompt: generatedPrompt, // Prompt for the image
    aspect_ratio: '16:9',
  };
}
```

## Node.js: Backend API Integration

This Node.js code manages communication between the frontend and the AI models (ChatGPT and Stable Diffusion). It processes the historical event, perspective, and emotion selected by the user, then queries ChatGPT to generate the narrative and visual prompts. It uses this data to generate images and send them back to the frontend.

# Result Showcase

MetaChronicle

Change perspective: Mid-age Businessman

Change emotion filter: Stressful

The Great Depression



Scene 1

*The city feels different today. As I step into my office, the once-bustling streets of New York seem quieter, almost suffocating. The news has already spread—the stock market has crashed. I can hear whispers from the halls. Investors are panicking. My stomach churns as I try to focus, but the weight of the situation presses on me. My fortune, my future, feels uncertain. The promise of prosperity now seems like a distant dream.*

Next

MetaChronicle

Change perspective: Mid-age Businessman

Change emotion filter: Stressful

The Great Depression



Scene 2

*By noon, the office is eerily quiet. I can't shake the feeling of impending doom. Clients have stopped calling, and those who do are pulling back their investments. There's less work to be done, and more time to worry. My colleagues, usually so composed, now share looks of concern. I'm trying to hold it together, but the stress is beginning to show. Every phone call, every email feels like a reminder that this crisis is far from over.*

Next

MetaChronicle

Change perspective: Mid-age Businessman

Change emotion filter: Stressful

The Great Depression



Scene 3

*The thought of losing everything keeps me awake at night. My home, my investments, my reputation—it's all slipping through my fingers. I lie awake in bed, staring at the ceiling, imagining the worst. The future looks bleak, and the weight of it all is unbearable. The noise of the city outside feels muffled, as if the whole world is closing in on me. The anxiety claws at my chest. What happens next?*

Next

MetaChronicle

Change perspective: Mid-age Businessman

Change emotion filter: Stressful

The Great Depression



Scene 4

*I try to push forward, but it's difficult to find hope. The streets are filled with people who look as lost as I feel. We all walk with our heads down, carrying our own burdens. I try to keep my business afloat, cutting costs and hoping that something will change. But with each passing day, the burden of worry only deepens. My confidence is slipping, replaced by a constant dread of failure.*

Next

MetaChronicle

Change perspective: Mid-age Businessman

Change emotion filter: Stressful

The Great Depression



Scene 5

*As weeks pass, the weight of the Great Depression still hangs over me. I take a moment on a quiet park bench, watching the city move around me, trying to find clarity. It's hard to escape the feeling of stress that has settled in my bones. But in this moment, I realize that the future may be uncertain, but there's still hope. I'm not alone in this struggle. Perhaps there's a way forward, even if I can't see it yet.*

Next

MetaChronicle

Change perspective: Mid-age Businessman

Change emotion filter: Stressful

The Great Depression

*What about experiencing the Great Depression through the eyes of a young, unemployed worker, where hope and resilience shine through despite the hardships?*

Try the Suggested Story

Return Home