A
Project Report
On
"AI VIDEO CREATION USING TREND ANALYSIS"

Prepared by

Mohit Patel(22DCS069)

Nir Patel(22DCS072)

Megha Joshi(D23DCS166)

Under the guidance of

Dr. Nilesh Dubey

Prof.Ankita Amaravat

Submitted to

Charotar University of Science & Technology

Degree of Bachelor of Technology

in Computer Science & Engineering

CSE305: Project-III

of 6th Semester of B.Tech

Submitted at

Computer Science and Engineering

Devang Patel institute Of Advance Technology and Research

Faculty of Technology & Engineering, CHARUSAT

At: Changa, Dist: Anand – 388421

November 2024

CHARUSAT
CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY

CERTIFICATE

This is to certify that the report entitled **"AI VIDEO CREATION USING TREND ANALYSIS"** is a Bonafide work carried out by **Mohit Patel (22DCS069)** under the guidance and supervision of **Prof. Ankita Amaravat** for the subject **Project-III (CSE305)** of 6th Semester of Bachelor of Technology in **DEPSTAR** at Faculty of Technology & Engineering – CHARUSAT, Gujarat.

To the best of my knowledge and belief, this work embodies the work of candidate himself, has duly been completed, and fulfills the requirement of the ordinance relating to the B.Tech. Degree of the University and is up to the standard in respect of content, presentation and language for being referred to the examiner.

Amaravat.

Prof. Ankita Amaravat
Assistant Professor
Dept.of Computer Science & Engineering
DEPSTAR, CHARUSAT-Changa.

Dr. Chirag Patel
Head,
Dept. of Computer Science & Engineering
DEPSTAR, CHARUSAT-Changa.

Devang Patel Institute of Advance Technology And Research

At: Changa, Ta. Petlad, Dist. Anand, PIN: 388 421. Gujarat

CHARUSAT
CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY

CERTIFICATE

This is to certify that the report entitled **"AI VIDEO CREATION USING TREND ANALYSIS"** is a bonafied work carried out by **Nir Patel (22DCS072)** under the guidance and supervision of **Prof. Ankita Amaravat** for the subject **Project-III (CSE305)** of 6th Semester of Bachelor of Technology in **DEPSTAR** at Faculty of Technology & Engineering – CHARUSAT, Gujarat.

To the best of my knowledge and belief, this work embodies the work of candidate himself, has duly been completed, and fulfills the requirement of the ordinance relating to the B.Tech. Degree of the University and is up to the standard in respect of content, presentation and language for being referred to the examiner.

Amaravat.

Prof. Ankita Amaravat
Assistant Professor
Dept.of Computer Science & Engineering
DEPSTAR, CHARUSAT-Changa.

Dr. Chirag Patel
Head,
Dept. of Computer Science & Engineering
DEPSTAR, CHARUSAT-Changa.

Devang Patel Institute of Advance Technology And Research

At: Changa, Ta. Petlad, Dist. Anand, PIN: 388 421. Gujarat

CHARUSAT
CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY

CERTIFICATE

This is to certify that the report entitled **"AI VIDEO CREATION USING TREND ANALYSIS"**is a Bonafide work carried out by **Megha Joshi (D23DCS166)** under the guidance and supervision of **Dr. Nilesh Dubey** for the subject **Project-III (CSE305)** of 6th Semester of Bachelor of Technology in **DEPSTAR** at Faculty of Technology & Engineering – CHARUSAT, Gujarat.

To the best of my knowledge and belief, this work embodies the work of candidate himself, has duly been completed, and fulfills the requirement of the ordinance relating to the B.Tech. Degree of the University and is up to the standard in respect of content, presentation and language for being referred to the examiner.

Dr.Nilesh Dubey
Assistant Professor
Dept.of Computer Science & Engineering
DEPSTAR, CHARUSAT-Changa.

Dr. Chirag Patel
Head,
Dept. of Computer Science & Engineering
DEPSTAR, CHARUSAT-Changa.

Devang Patel Institute of Advance Technology And Research

At: Changa, Ta. Petlad, Dist. Anand, PIN: 388 421. Gujarat

DECLARATION BY THE CANDIDATES

We hereby declare that the project report entitled "**AI VIDEO CREATION USING TREND ANALYSIS**" submitted by us to Devang Patel Institute Of Advance Technology and Research, Changa in partial fulfilment of the requirements for the award of the degree of **B. Tech Computer Science & Engineering**, from the Department of Computer Science & Engineering, DEPSTAR, FTE is a record of Bonafide CSE305 Project-III carried out by us under the guidance of **Dr. Nilesh Dubey and Prof. Ankita Amaravat**. We further declare that the work carried out and documented in this project report has not been submitted anywhere else either in part or in full and it is the original work, for the award of any other degree or diploma in this institute or any other institute or university.

Signature of the candidate
(Mohit Patel -22DCS069)

Signature of the candidate
(Nir Patel -22DCS072)

Signature of the candidate
(Megha Joshi-D23DCS166)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Dr. Nilesh Dubey
Assistant Professor
Department of Computer Science & Engineering,
Devang Patel Institute of Advance Technology and Research
(DEPSTAR)
Faculty of Technology (FTE)
Charotar University of Science and Technology (CHARUSAT) - Changa.

Prof. Ankita Amaravat
Assistant Professor
Department of Computer Science & Engineering,
Devang Patel Institute of Advance Technology and Research (DEPSTAR)
Faculty of Technology (FTE)
Charotar University of Science and Technology (CHARUSAT) - Changa.

ABSTRACT

The project "AI Video Creation using Trend Analysis" introduces an innovative end-to-end automated pipeline designed to revolutionize content creation, especially for news content creators and social media influencers. The proposed system eliminates the traditional manual processes of scraping web data, scriptwriting, and video shooting by providing a single integrated platform to automate the entire video creation pipeline.
This project leverages multiple advanced technologies and APIs, including NewsAPI for trend fetching, DuckDuckGo for contextual news collection, VADER for sentiment analysis, Groq API integrated with the LLaMA model for script generation, and Wav2Lip for video creation through image-based lip-syncing.

The complete pipeline begins with the detection of trending topics from NewsAPI. Upon selecting a desired trend, relevant news articles are fetched automatically. These articles undergo sentiment analysis using the VADER model to generate an emotionally resonant perspective. The processed data, including news context and sentiment, is passed to the Groq API, using the LLaMA model, to generate a natural 3-minute conversational script suitable for video narration. The generated script is further converted into an audio file using text-to-speech models. Subsequently, the user uploads a static image along with the generated audio, and Wav2Lip is employed to create a lip-synced video, generating realistic facial movements in synchronization with the audio.

This AI-driven approach transforms the conventional video content creation workflow into a fully automated process, enabling faster, accurate, and scalable video generation from trending content, thus helping creators stay ahead in the dynamic digital content landscape.

TABLE OF CONTENT

LIST OF FIGURES:

CHAPTER 1: INTRODUCTION

1.1 Project Overview

- The rise of social media and digital content consumption has significantly increased the demand for fast, accurate, and engaging video content. The traditional video content creation process is tedious and time-consuming, involving content research, scriptwriting, audio recording, video shooting, and editing.
- This project presents *"AI Video Creation using Trend Analysis"*, an integrated automated platform that helps users generate videos based on trending topics without any manual effort. It provides a complete solution — from trend detection to final video generation — thereby transforming the video creation landscape.

1.2 Objectives

- Automate the detection of trending topics using NewsAPI..
- Fetch related news content using DuckDuckGo.
- Perform sentiment analysis on the news data using VADER.
- Generate video scripts using Groq API and LLaMA models.
- Convert scripts into audio using Text-to-Speech tools.
- Create lip-synced videos from an image and audio using Wav2Lip.

1.3 Scope

- Automating the entire video content creation pipeline.
- Providing an interface for users to select trending topics.
- Manage project files and export them easily.
- Use customizable templates.
- Run lightweight models for offline or faster inference.

1.4 Summary of Key Outcomes and Interpretations

Category	Technology
Trend Detection	NewsAPI, DuckDuckGo
Sentiment Analysis	VADER
Script Generation	Groq API + LLaMA Model
Audio Generation	Text-to-Speech (Edge TTS)
Video Generation	Wav2Lip
Backend	Python, Flask
Frontend	Streamlit / Flask UI
Deployment	Local and Cloud-based

CHAPTER 2: OVERVIEW OF EXISTING SOLUTION/TECHNOLOGY/METHODOLOGY

2.1 Existing Low-Code/No-Code Platforms

There are several platforms that aim to simplify web development through drag-and-drop interfaces and pre-built components. Examples include Wix, Webflow, and Bubble.io. While they offer ease of use, they often:

- Lack flexibility in code customization.
- Do not support direct natural language input.
- Are limited in handling complex or responsive layouts without manual tweaks.

These platforms are visual but not intelligent—they require users to adjust UI manually, and they do not generate or understand underlying HTML/CSS from user intent.

2.2 Code Generation Using AI

With the rise of Large Language Models (LLMs) like OpenAI's Codex, Google's Gemini, and Meta's LLaMA, AI code generation has become more accessible. These models can generate code from natural language prompts. However:

- Most are integrated into IDEs like GitHub Copilot, which still require coding knowledge.
- They lack structure enforcement for responsive design or template layout continuity.
- They are not integrated into an end-to-end platform with live preview or file management.

2.3 Limitations in Current Methodologies

Current solutions either:

- Focus too much on UI drag-and-drop (no intelligence),
- Or rely on AI in isolated environments (not part of a platform).

There’s a gap between AI-generated code and real-time web development tools. Users either have to manually edit code, struggle with integration, or use tools that don’t fully understand prompt context.

CHAPTER 3: LIMITATIONS OF EXISTING SYSTEM

Despite the availability of several low-code and no-code platforms, most existing systems suffer from key limitations that hinder the efficiency and flexibility of modern web development. Traditional platforms like Wix, Webflow, and Bubble provide visual editors with drag-and-drop functionality, but they lack the intelligence to interpret user intent from natural language. These tools often limit users to predefined components, restricting customization and creative freedom. Furthermore, while they simplify front-end design, they don't offer fine control over the generated code or responsive behaviors, making it difficult for developers to scale or extend projects beyond basic layouts.

On the other hand, AI-powered code generation tools such as GitHub Copilot or OpenAI's Codex are typically integrated into development environments and still require programming knowledge. These tools are prompt-based but are not part of a complete web-building ecosystem—they do not support live previews, template integration, or project-level file management. Moreover, they often generate code in isolation without maintaining the structural or design consistency of an ongoing web project. There is also no direct support for real-time editing, collaboration, or deploying web projects directly from these AI assistants.

Overall, the current systems either focus on ease of use with limited intelligence or provide intelligent assistance without accessibility for non-technical users. This results in a fragmented workflow where users must switch between tools to design, code, preview, and export. Project Bolt aims to overcome these limitations by unifying the development experience through AI-driven natural language coding, real-time editing, and full project lifecycle support in one platform.

CHAPTER 4: NEED FOR NEW SYSTEM / ALGORITHM

The rapid evolution of AI and the growing demand for faster web development highlight the need for a smarter, more integrated system that goes beyond traditional drag-and-drop builders or isolated AI code tools. Existing platforms either cater to beginners with limited functionality or to developers who already possess coding skills. There is a clear gap for a unified solution that can **translate natural language into structured, editable website code** while maintaining responsiveness and design integrity.

Project Bolt introduces a novel system that combines the power of large language models—such as **GROQ's LLaMA 3.3 70B**—with an intelligent development environment. This new approach enables users to describe their design intentions in plain English and instantly receive valid HTML/CSS output, which they can view and edit in real-time. Additionally, the use of a **fine-tuned Gemma 3 1B model with 4-bit LoRA** provides a lightweight, cost-effective alternative for edge deployments, ensuring accessibility and performance even on limited hardware.

By integrating prompt-based AI generation, live editing, file management, and export features into one seamless platform, Project Bolt meets the needs of both novice users and experienced developers. It presents a significant shift in how websites can be created—**intelligently, efficiently, and with minimal technical barriers**.

CHAPTER 5: SYSTEM DESIGN

5.1 System Architecture Overview

Project follows a modular, layered architecture that integrates frontend, backend, AI model APIs, and a file system to deliver an end-to-end intelligent web development experience. The platform is divided into four key layers: User Interface Layer, Backend API Layer, AI Model Integration Layer, and Project Management Layer. This architecture enables seamless interaction between the user and the system while ensuring scalability, modularity, and performance.

5.2 Frontend Design

The frontend is built using React.js, TypeScript, and Tailwind CSS to deliver a responsive, Interactive user interface. Key components include:

- A Monaco-based code editor for real-time HTML/CSS editing.
- A live preview panel that instantly reflects code updates.
- A project explorer sidebar for file and folder management.
- A prompt input field where users describe their desired layout in natural language.

All frontend components communicate asynchronously with the backend using RESTful APIs.

5.3 Backend Design

The backend is developed using Python with either FastAPI or Flask, responsible for:

- Handling API requests and user prompts.
- Routing requests to appropriate AI models.
- Managing session states, templates, and file data.
- Returning generated or updated code to the frontend.

It also ensures secure, efficient communication between the frontend and AI models

5.4 AI Model Integration Layer

This layer connects the backend with two key AI models:

- LLaMA 3.3 70B (via GROQ): Used for accurate, high-quality HTML/CSS generation from natural language prompts.
- Gemma 3 1B (LoRA 4-bit): A fine-tuned lightweight model for edge deployment, ideal for faster or offline code generation.

Prompts are sent to the selected model, and the returned HTML/CSS code is sent back to the user through the backend.

5.5 Project Management Layer

This layer handles:

- File creation, deletion, and updates.
- Template loading and customization.
- Project export (as downloadable ZIP files or Git integration).
- Future features like user accounts, project versioning, and collaboration

5.6 Data Flow

- User enters a prompt in the UI.
- Frontend sends the prompt to the backend via API.
- Backend routes the prompt to the selected AI model.
- Model returns HTML/CSS code based on the prompt.
- Code is rendered in the editor and live preview.
- Users can edit, manage files, and export projects.

CHAPTER 6: IMPLEMENTATION

6.1 Frontend Implementation

The frontend of Project Bolt is developed using React.js, TypeScript, and Tailwind CSS, ensuring a responsive and dynamic user interface. Key components implemented include:

- Prompt Input Panel: Allows users to enter natural language prompts describing the desired web layout or component.
- Live Preview Pane: Automatically updates to reflect code changes, providing instant visual feedback.
- Project Explorer: Lets users create, rename, delete, and manage project files and folders in a structured way.

State management is handled using React's Context API and hooks for efficient UI updates.

6.2 Backend Implementation

The backend is built using Python with FastAPI, chosen for its speed and simplicity in building RESTful APIs. Backend functionalities include:

- Receiving prompt data from the frontend.
- Handling API requests to call the appropriate AI model (LLaMA 3.3 70B or Gemma 3 1B).
- Returning generated code and managing file structure.
- Performing operations like saving, exporting, and organizing user projects.

All APIs are structured modularly, making the backend easy to scale and maintain.

6.3 AI Model Integration

Two AI models are used in implementation:

- LLaMA 3.3 70B (via GROQ API): Used for high-accuracy, server-side code generation.
- Gemma 3 1B (4-bit LoRA): Deployed locally after fine-tuning for lightweight, fast HTML/CSS code generation.

A model selection toggle was implemented, allowing the system to switch based on user choice or hardware availability.

6.4 File Management & Export

A custom file manager was developed that mimics the structure of IDEs. Users can:

- Create multiple HTML/CSS/JS files.
- Organize them into folders.
- Download the entire project as a .zip file for deployment or editing elsewhere.

6.5 Deployment and Testing

- The platform was tested locally and deployed using Vercel for the frontend and Render/Railway for backend services.
- The AI inference pipeline for the Gemma model was tested on a GPU-enabled local environment.
- Integration testing ensured smooth communication between frontend, backend, and AI APIs.

CHAPTER 7: RESULTS & DISCUSSIONS

7.1 Result

The implementation of Project Bolt successfully achieved the goal of enabling users to generate responsive website components using natural language instructions. The integration of the LLaMA 3.3 70B model (via GROQ) produced highly accurate and clean HTML/CSS code across a wide variety of prompt inputs. The real-time preview, Monaco editor, and project file manager provided a complete development environment that significantly reduced the time required to build web pages.

In parallel, the fine-tuned Gemma 3 1B (LoRA 4-bit) model delivered promising results. While slightly less accurate than LLaMA, it demonstrated good performance in generating basic layouts and components, making it a viable solution for edge devices or local development.

The following outcomes were observed:

- Prompt-to-code conversion accuracy (LLaMA): ~93% for common components like navbars, cards, and forms.
- Average generation time: <1.5s (LLaMA), ~0.6s (Gemma) on local setup.
- User experience feedback: Smooth workflow and intuitive interface for both technical and non-technical users.

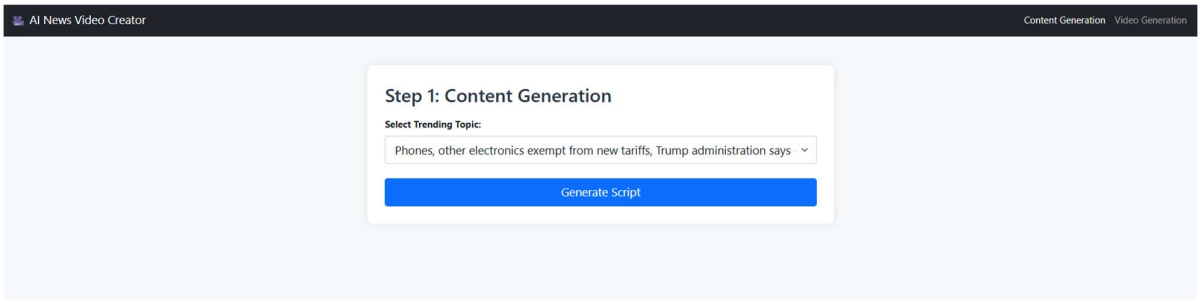

Figure 1:Trend Fetching

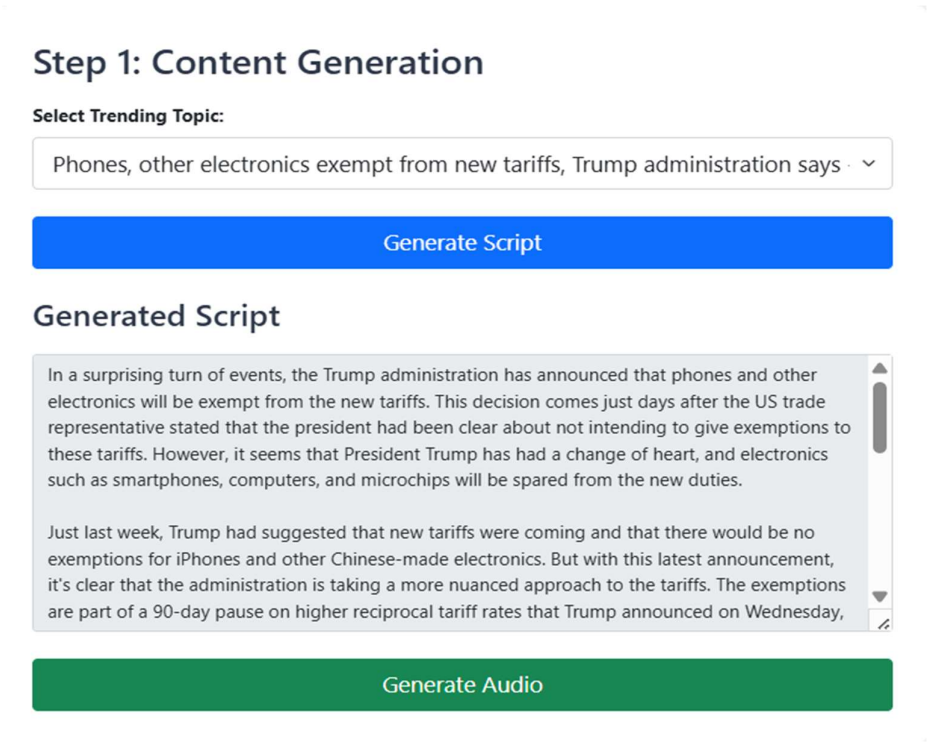

Figure 2 : Script Generated

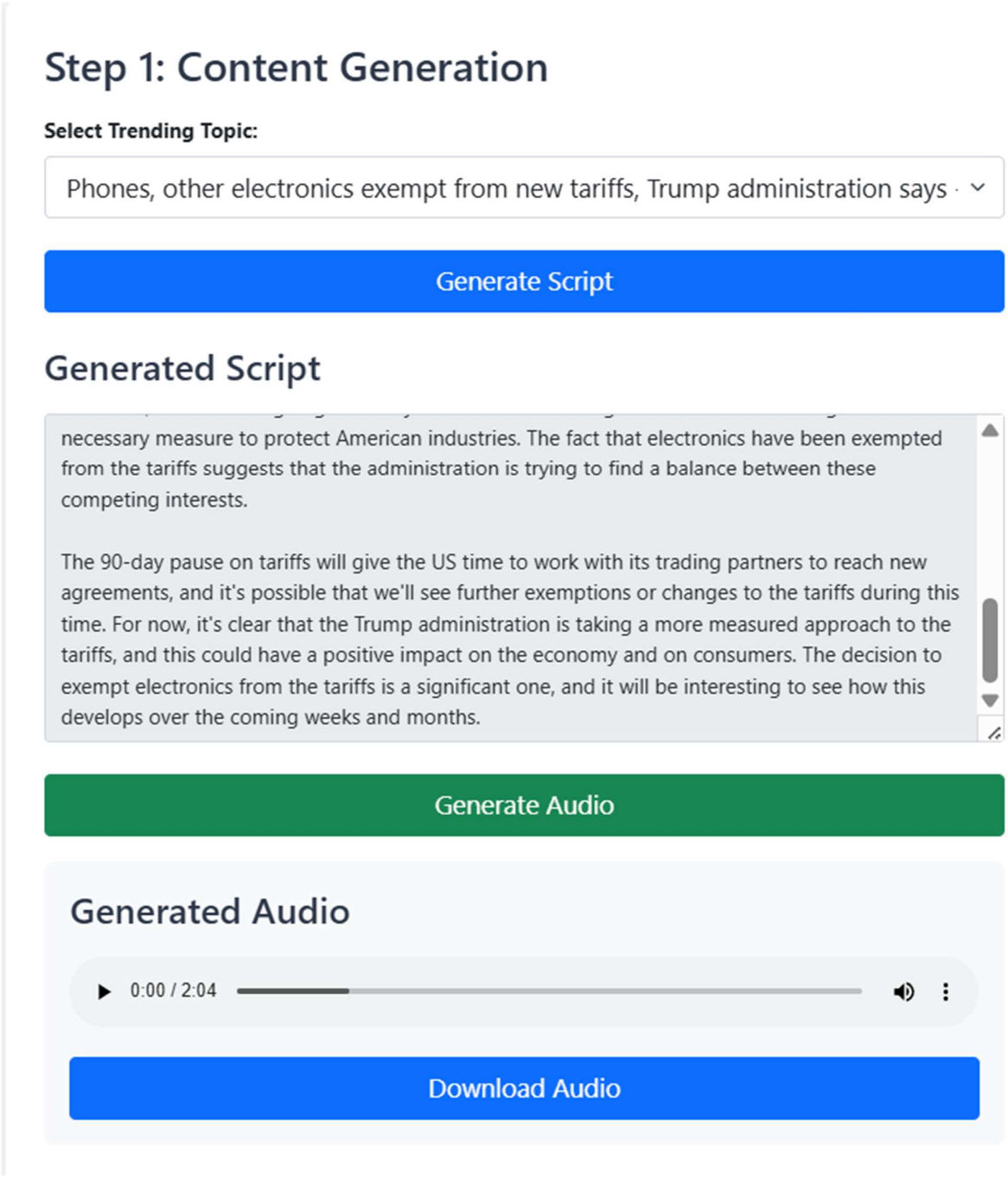

Figure 3 : Audio Generation

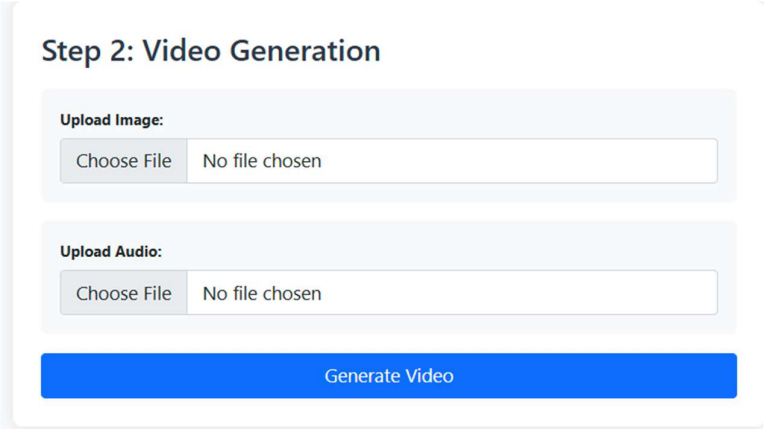

Figure 4 : Inputs for Video Generation

7.2 Discussions

The results indicate that integrating a powerful LLM into a low-code/no-code platform can dramatically simplify the website creation process. The system-maintained template consistency, responsive design principles, and semantic structure, which are critical for modern web development.

However, limitations were also noted. The Gemma model, though fast and lightweight, occasionally generated incomplete or incorrect tags for complex layouts. This can be improved with additional training data or better preprocessing. Additionally, the live preview system had minor rendering delays during frequent code edits, which could be optimized with improved state handling or virtualization techniques.

Overall, Project Bolt demonstrates a successful fusion of AI and software engineering, offering a robust prototype for the future of intelligent, accessible web development.

CHAPTER 8: CONCLUSION

Project successfully delivers an innovative AI-powered low-code/no-code web development platform that allows users to create responsive websites through natural language prompts. By integrating advanced language models like GROQ's LLaMA 3.3 70B for high-quality code generation and the fine-tuned Gemma 3 1B (4-bit LoRA) for lightweight inference, the system bridges the gap between non-developers and web development.

The platform's modern tech stack—React.js, TypeScript, Tailwind CSS, and FastAPI—enables a seamless and real-time development experience through its Monaco editor, live preview, and file management features. The combination of intuitive UI/UX and intelligent backend processing ensures that users can generate, edit, and export fully functional web projects without needing deep coding knowledge.

In conclusion, Project Bolt not only showcases the practical application of large language models in real-world software but also sets the foundation for future advancements in AI-assisted development. It simplifies the website creation process, reduces development time, and empowers users of all skill levels to build high-quality web solutions with ease.

CHAPTER 9: REFERENCES

1. Bolt.new: AI-Powered Full-Stack Web Development (https://bolt.new/)

2. Introducing Llama 3.3 70B Speculative Decoding on GroqCloud (https://groq.com/groq-first-generation-14nm-chip-just-got-a-6x-speed-boost-introducing-llama-3-1-70b-speculative-decoding-on-groqcloud/)

3. Fine-Tune Gemma using Hugging Face Transformers and QLoRA (https://ai.google.dev/gemma/docs/core/huggingface_text_finetune_qlora)

4. Monaco Editor React Integration (https://www.npmjs.com/package/@monaco-editor/react)

5. Flask vs FastAPI Comparison (https://www.reddit.com/r/FastAPI/comments/1dl7fp2/flask_vs_fastapi/)

6. Responsive Design with Tailwind CSS (https://tailwindcss.com/docs/responsive-design)

7. Using TypeScript with React (https://react.dev/learn/typescript)

8. AI Code Generation Use Cases (https://cloud.google.com/use-cases/ai-code-generation)

9. Low-Code vs. No-Code Development Platforms (https://www.microsoft.com/en-us/power-platform/products/power-apps/topics/low-code-no-code/low-code-no-code-development-platforms)

10. StackBlitz Bolt.new GitHub Repository (https://github.com/stackblitz/bolt.new)