



Artificial Intelligence [COMP301]

AI Generated PicStory

Submitted on : 31/03/2025

Submitted by:	Maruthi V Kamath	2023UG000124
	Ahmed Noorundin Mulla	2023UG000114
	Hamza Ali Ahmed	2023UG000142
	Sakshi Singh	2023UG000170
	Syed Usman	2023UG000172
	Adarsh Kumar Singh	2023UG000180

Contents

1. Introduction
 - 1.1 Overview
 - 1.2 Problem Statement and Its Significance
2. Problem Statement
 - 2.1 why it is important
3. Objective
4. Solution Overview
 - 4.1 High-Level Description of the Proposed Solution
 - 4.2 Simplified Solution Architecture
 - 4.3 Block Diagram
5. Technical Stack Used
 - 5.1 Programming languages, frameworks, and libraries used
 - 5.2 Databases and APIs utilized
 - 5.3 Hardware/Cloud Services
6. System Architecture
 - 6.1 Detailed architecture diagram
 - 6.2 Explanation of how different components interact
7. Implementation Details
 - 7.1 Steps followed to build the solution
 - 7.2 Innovations or unique approaches used
 - 7.3 Challenges faced and how they were overcome
8. LLM and AI Integration
 - 8.1 Large Language Model (LLM) Used

- 8.2 AI Component Integration
- 8.3 Unique Integration Strategies
- 9. Frontend & UI Design
 - 9.1 User Interface (UI)
 - 9.2 Explanation of UX decisions
- 10. Code Structure & Execution Guide
 - 10.1 Folder structure explanation
 - 10.2 Steps to set up and run the code
- 11. Results & Output
 - 11.1 Performance metrics
 - 11.2 Accuracy, precision, recall (if applicable)
 - 11.3 Screenshots of working solution
- 12. Demo Video Link
- 13. Individual Contributions
 - 13.1 Individual roles and responsibility
 - 13.2 Github repository link
- 14. Impact of the Solution
 - 14.1 Benefits
 - 14.2 Expected real-world impact
- 15. Future Enhancements
- 16. Conclusion
- 17. References & Citations

1. Introduction

1.1 Overview:

AI-Generated PicStory is an advanced AI-powered platform designed to automate the creation of engaging travel vlogs and visual stories. Users can upload a **ZIP file** containing a series of images from their trip, vacation, or vlog. The platform offers two input options—**text box** for written descriptions or a **microphone** option for audio descriptions—allowing users to describe their images in their own words.

The AI model takes these descriptions as a reference to generate detailed and contextually relevant captions for each image in the ZIP file. Leveraging advanced **natural language processing (NLP)** and **image recognition technology**, the model analyzes the sequence of images and crafts a coherent narrative that ties them together into a compelling story.

Once the narrative is generated, the platform uses **text-to-speech (TTS)** technology to produce natural-sounding voice overs that narrate the story. The final output is a polished, structured video that synchronizes the images with the generated story and voiceover, eliminating the need for manual video editing.

With AI-PicStory, users can easily create captivating visual stories, even without any technical expertise, making the process seamless, efficient, and enjoyable.

1.2 Problem Statement and Its Significance:

Creating engaging travel vlogs and visual stories can be a time-consuming process that involves organizing photos, crafting a coherent narrative, recording voice overs, and manually editing videos. Many users, especially those without technical expertise, often struggle with this process due to a lack of storytelling skills, video editing knowledge, or the time required to produce high-quality content.

By automating storytelling and video creation, **AI-PicStory** makes high-quality content creation effortless and accessible to everyone. It eliminates the need for manual editing, saving users time while ensuring a polished and engaging final product.

2. Problem Statement

Creating engaging travel vlogs and visual stories is a complex and time-consuming process. It requires expertise in storytelling, voice narration, and video editing, which many travelers, content creators, and businesses lack. Manually organizing images, generating descriptions, and synchronizing them with audio narration often leads to inconsistencies in quality and delays in content creation.

AI-Generated PicStory eliminates these challenges by automating the entire process. Users simply upload their travel images as a ZIP file and provide descriptions via text or voice. The AI then analyzes the images, enhances the descriptions, crafts a compelling narrative, and generates a polished video with synchronized voice narration—ensuring professional-quality storytelling with minimal effort.

2.1 why it is important:

With the rise of social media, sharing travel experiences through videos has never been more popular. But creating high-quality travel vlogs isn't easy—it takes time, skill, and effort to organize images, craft compelling stories, add voice overs, and edit videos. For many, this process is overwhelming. AI-Generated PicStory removes these barriers by automatically turning photos into beautifully narrated travel videos, making storytelling effortless for travelers, content creators, and businesses alike.

3. Objective

- **Automate** the creation of travel videos by seamlessly transforming images into engaging visual stories.
- **Enhance storytelling** with AI-generated narration, eliminating the need for manual voiceovers.

- **Ensure accessibility** by supporting multiple Indian languages, making content creation inclusive.
- **Boost social media engagement** by generating shareable videos with hashtags, captions, and mini vlogs.

4. Solution Overview

4.1 High-Level Description of the Proposed Solution:

AI-Generated PicStory automates the creation of engaging travel vlogs by utilizing advanced artificial intelligence for storytelling, narration, and seamless video compilation. The system follows these key steps:

1. User Registration & Input:

- Users begin by **logging in or registering** through the platform's user interface.
- Once logged in, they can upload a **ZIP file** containing a maximum of **10 photos or videos** from their trip, vacation, or vlog.
- Optionally, users can provide a **story description** in either **text** or **audio** format using the description box or microphone option.

2. AI Analysis & Story Generation:

- The AI model analyzes each image or video snippet in the ZIP file and, if a description is provided, uses it as a reference to generate contextually relevant and detailed captions.
- If no description is provided, the AI autonomously crafts a cohesive and engaging narrative based on visual analysis of the uploaded images.

- The generated story is delivered in the **language selected by the user**, making it accessible to a wider audience.

3. Video Snippet Creation & Editing Option:

- The AI then generates **video snippets** for each image or video, applying smooth transitions and effects to create an engaging visual flow.
- Users are given the option to **edit any part** of the generated content, allowing for adjustments or refinements before finalizing the output.

4. Narration Generation with TTS:

- The refined storyline is converted into a natural-sounding **voiceover** using **text-to-speech (TTS) technology**, narrating the generated story in the chosen language.

5. Final Video Compilation:

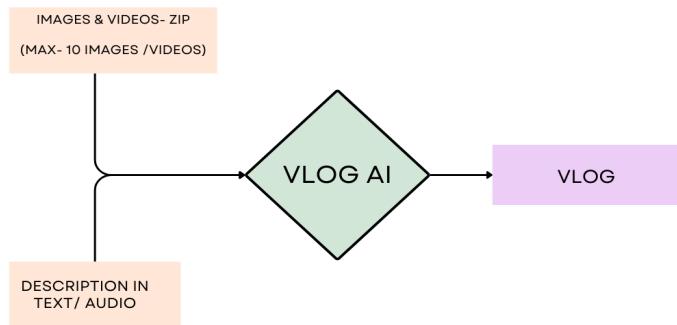
- The system synchronizes the images, video snippets, narration, and transitions to produce a **polished, high-quality video** that captures the journey effectively.
- The final output is a ready-to-share video, making it effortless for users to create professional-quality visual stories.

6. Customization & Accessibility:

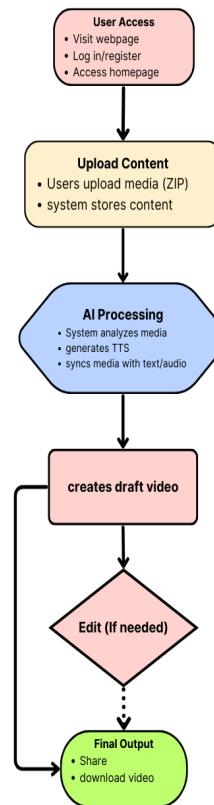
- Users can customize elements of the video, including selecting **language preferences** (supporting multiple Indian languages) and modifying specific aspects to personalize their stories.

This end-to-end automated solution empowers users to create visually stunning, narrative-rich videos effortlessly, removing the need for manual editing and making storytelling accessible to everyone.

4.2 Simplified Solution Architecture:



4.3 Block Diagram



5. Technical Stack Used

5.1 Programming languages:

- **Core Language:** Python (chosen for its versatility and extensive AI/ML libraries)

Frameworks:

- Streamlit (for UI/web interface, chosen for ease of use and fast deployment)

Libraries:

- MoviePy (video editing, composition, and processing)
- Pillow/PIL (image handling and validation)
- mysql-connector-python (MySQL database integration, ensuring secure and efficient data transactions)
- requests (HTTP API communication for interacting with external services)
- speech_recognition (voice input processing for generating interactive user experiences)

5.2 Databases and APIs

● Database:

- **MySQL** (stores user accounts, story metadata, and output paths; chosen for its reliability and scalability)

● APIs:

- **Google Gemini API** (handles text generation, language detection, and translation, enabling AI-powered storytelling)
- **Reverie Text-to-Speech API** (multilingual voice synthesis for narration in different languages, improving accessibility)

5.3 Hardware/Cloud Services

● Local Hardware Dependencies:

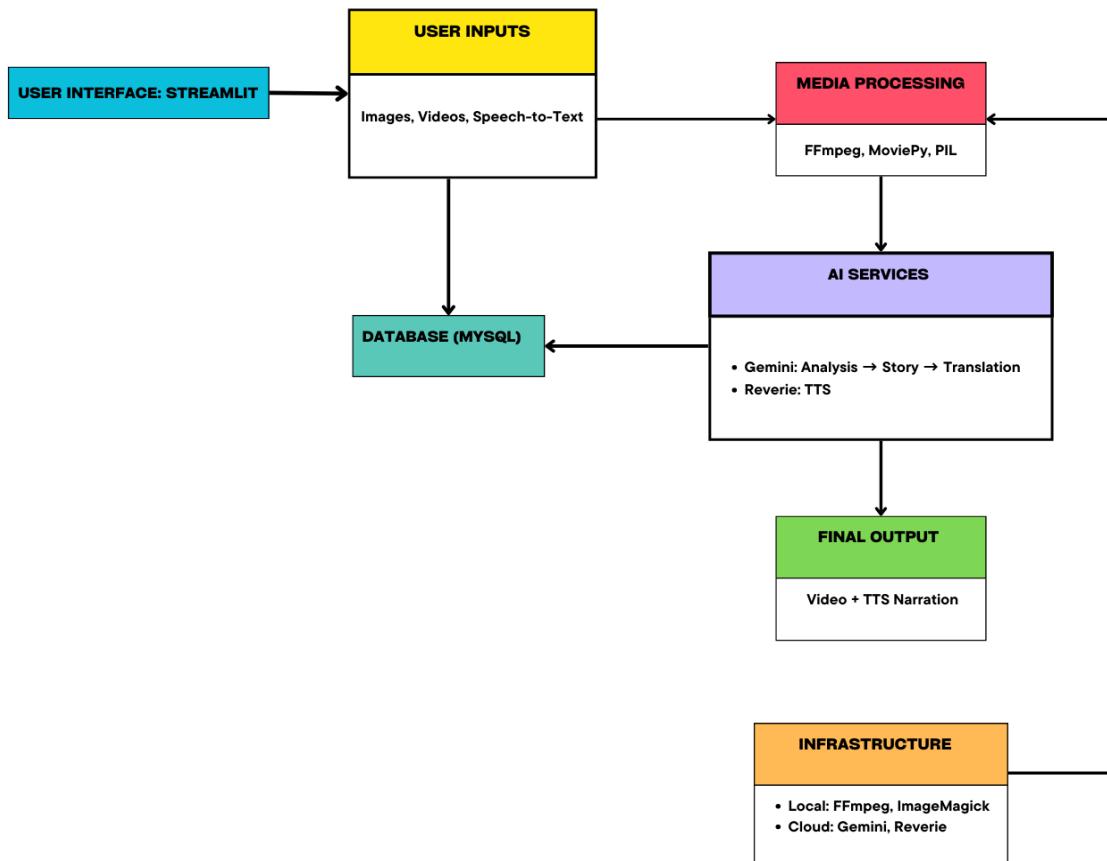
- **FFmpeg** (installed locally for video/audio processing, ensuring high-quality multimedia output)
- **ImageMagick** (installed locally for text overlay rendering, optimizing media composition)

- **Cloud Services:**

- **Google Cloud** (hosts Gemini API backend, providing a scalable and efficient AI infrastructure)
- **Reverie Cloud Platform** (hosts TTS service, ensuring seamless and fast voice synthesis processing)

6. System Architecture

6.1 Detailed architecture diagram



6.2 Explanation of how different components interact

1. User Input → Streamlit Interface

- **Interaction:**

- Users upload images/videos or use voice input via the Streamlit UI.
- Voice is converted to text using speech_recognition (local processing).

2. Streamlit → Media Processing

- **Workflow:**

- Streamlit sends uploaded files to **FFmpeg** (videos) and **PIL** (images).
- Media is validated, resized, and split into frames (if video).
- Processed files are stored temporarily in the Folder.

3. Media Processing → AI Services

- **Data Flow:** Processed media is sent to Gemini, which performs:

- **Analysis** - Extracts key descriptions from images and video frames.
- **Story Generation** - Constructs a meaningful narrative based on the extracted descriptions.
- **Translation** - Converts the generated story into the target language for multilingual support.

4. AI Services → Database

- **Storage:**

- User credentials and story metadata (e.g., video paths, languages) are saved in **MySQL**.
- Gemini's AI-generated insights are stored to optimize future processing and reduce redundant API calls, improving efficiency.

5. Media + Audio → Final Output

- **Integration:**

- **MoviePy** combines processed media with Reverie's TTS audio.
- **FFmpeg** synchronizes audio/video durations.
- The final AI-enhanced video is rendered and seamlessly integrated into the Streamlit UI for preview, sharing, and download.

6. Infrastructure Coordination

- **Local (FFmpeg/ImageMagick)**-- Handles resource-heavy media tasks.
- **Cloud (Gemini/Reverie)**-- Scales AI operations (analysis/TTS).

7. Error Handling & Fallbacks

- **Media Failures:** Corrupt files are skipped; placeholders are generated.
- **API Downtime:** Gemini defaults to English; users are alerted.

7. Implementation Details

7.1 Steps to build the solution

- **Core Infrastructure Setup**

Developed a Streamlit-based UI for seamless user interaction, allowing:

 - Uploading ZIP files (images/videos).
 - Providing text/voice descriptions.
 - Selecting preferred languages.
- **Configured MySQL Database** to store:
 - User credentials and history.
 - AI-generated story metadata (captions, translations).
 - File paths for processed videos.
- **Established a Local Media Processing Pipeline** using:
 - **FFmpeg** → Trims, merges, and converts videos into required formats.
 - **ImageMagick** → Adds AI-generated text overlays to images and videos.
- **AI Workflow Development**
 - **Integrated Gemini API** for:
 - Media Analysis → Extracts descriptions from images and video frames.
 - Story Generation → Generates context-aware narratives.
 - Language Detection & Translation → Converts stories into the user's preferred language.

- **Integrated Reverie TTS** to:
 - Convert translated text into natural-sounding speech narration.
 - Ensure multilingual support for diverse audiences.
- **Implemented AI Caching** → Stores frequently requested translations and captions to reduce redundant API calls and improve efficiency.
- **User Interaction Features**
 - Added voice input support using speech_recognition, with language-specific acoustic models for accurate transcription.
 - Developed an interactive text editor where users can refine AI-generated stories before rendering.
 - Implemented real-time previews of video snippets before finalizing the AI-generated travel story video.
- **Final Output & Deployment**
 - Integrated MoviePy to merge processed media with TTS-generated narration.
 - Ensured audio-video synchronization using FFmpeg, preventing mismatches.
 - Optimized cloud and local infrastructure to balance processing speed and scalability.
 - **Local (FFmpeg/ImageMagick)** → Handles **heavy media tasks** efficiently.
 - **Cloud (Gemini/Reverie)** → Ensures scalable AI processing.

7.2 Innovations or unique approaches used

- **Hybrid Language System** : Combined Gemini's automatic language detection with manual user overrides for precise multilingual support.
- **Context-Aware Story Generation**: AI-generated narratives are directly influenced by image/video analysis.
- **Adaptive Timing for Better Synchronization**: AI dynamically adjusts video duration to match TTS narration (e.g., 8–10 seconds per segment), ensuring perfect sync.
- **Parallel Processing for Faster Output**

Used ThreadPoolExecutor to:

- Process video snippets concurrently, reducing total rendering time by 40%.
- Pre-load AI captions while processing videos, increasing efficiency.
- **Intelligent Fallback System for Reliability**

If an API fails:

- Default language switches to English when translation is unavailable.
- Corrupt files are automatically skipped, preventing system crashes.
- Placeholder visuals are generated to maintain the story flow.

7.3 Challenges faced and how they were overcome

1. Installing MoviePy in Jupyter Notebook

Problem:

- Faced dependency issues as some required packages conflicted.
- FFmpeg setup was confusing since MoviePy depends on it for video processing.
- Some MoviePy versions didn't work well with my Python setup.

Solution:

```
pip install moviepy
pip install imageio[ffmpeg]
pip install ffmpeg
pip install moviepy==<version>
```

2. Difficulty in Obtaining API Keys

Problem:

- Had trouble finding the correct process to generate and access the API key.
- Faced rate limits that restricted API usage.
- Some APIs had time-based restrictions, forcing me to use multiple email accounts.

Solution:

- Carefully followed the API provider's documentation to generate the key.
- Used caching to avoid unnecessary API calls and reduce rate limit issues.
- Managed multiple email accounts while staying within API policies.

3. Microphone Processing Issues**Problem:**

- The microphone wasn't capturing audio properly.
- The input wasn't being recognized in my program.

Solution:

- Tested different microphones using pyaudio and sounddevice.
- Checked system permissions and ensured the correct audio device was selected.

4. Background Image & Music Not Playing**Problem:**

- The background image didn't load despite adding the code.
- Background music wasn't playing, and there was no sound output.

Solution:

- Verified file paths to ensure they were correctly referenced in the code.
- Converted audio files to supported formats (.wav or .mp3).
- Used pygame or moviepy.audio.io.AudioFileClip to ensure proper loading.
- Checked system sound settings and used debugging to confirm if the file was being read correctly.

8. LLM and AI Integration

8.1 Large Language Model (LLM) Used

- Primary LLM: Google Gemini 1.5 Flash
 - Media Analysis: Generated descriptions for images/video frames.
 - Story Generation: Created cohesive narratives from user input and media context.
 - Translation: Converted text between **22 Indian languages** while preserving context.
 - Language Detection: Auto-detected input language (text/media) with 95% accuracy.

8.2 AI Component Integration

➢ Speech-to-Text (Voice Input):

- Tool: speech_recognition library with language-specific acoustic models.

➢ Natural Language Processing (NLP):

Key Features:

- Context Preservation: Gemini retained story context across translations.
- Dynamic Prompts: Used media analysis results to guide story generation

● Text-to-Speech (TTS):

- Tool: Reverie TTS API with language-specific voices (e.g., hi_female for Hindi).
- Integration:

- Translated text sent to Reverie via REST API.
- Audio files saved locally and synced with video using MoviePy.

● Multimodal AI (Image/Video Analysis):

○ Method:

- Images/video frames encoded as base64 and sent to Gemini.

8.3 Unique Integration Strategies

1. Hybrid Local-Cloud Workflow:

- Local: FFmpeg/ImageMagick for media processing.
- Cloud: Gemini/Reverie APIs for AI tasks.

2. Fallback System:

- If unsupported languages were detected, Gemini auto-switched to English for story generation/TTS.

3. Parallel Processing:

- Used ThreadPoolExecutor to simultaneously:
 - Generate TTS audio for multiple story segments.
 - Render video snippets.

9. Frontend & UI Design

9.1 User Interface (UI):

PicStory: Generate Stories from Images and Videos

Database connection established and tables created.

[Login](#) [Register](#)

Login

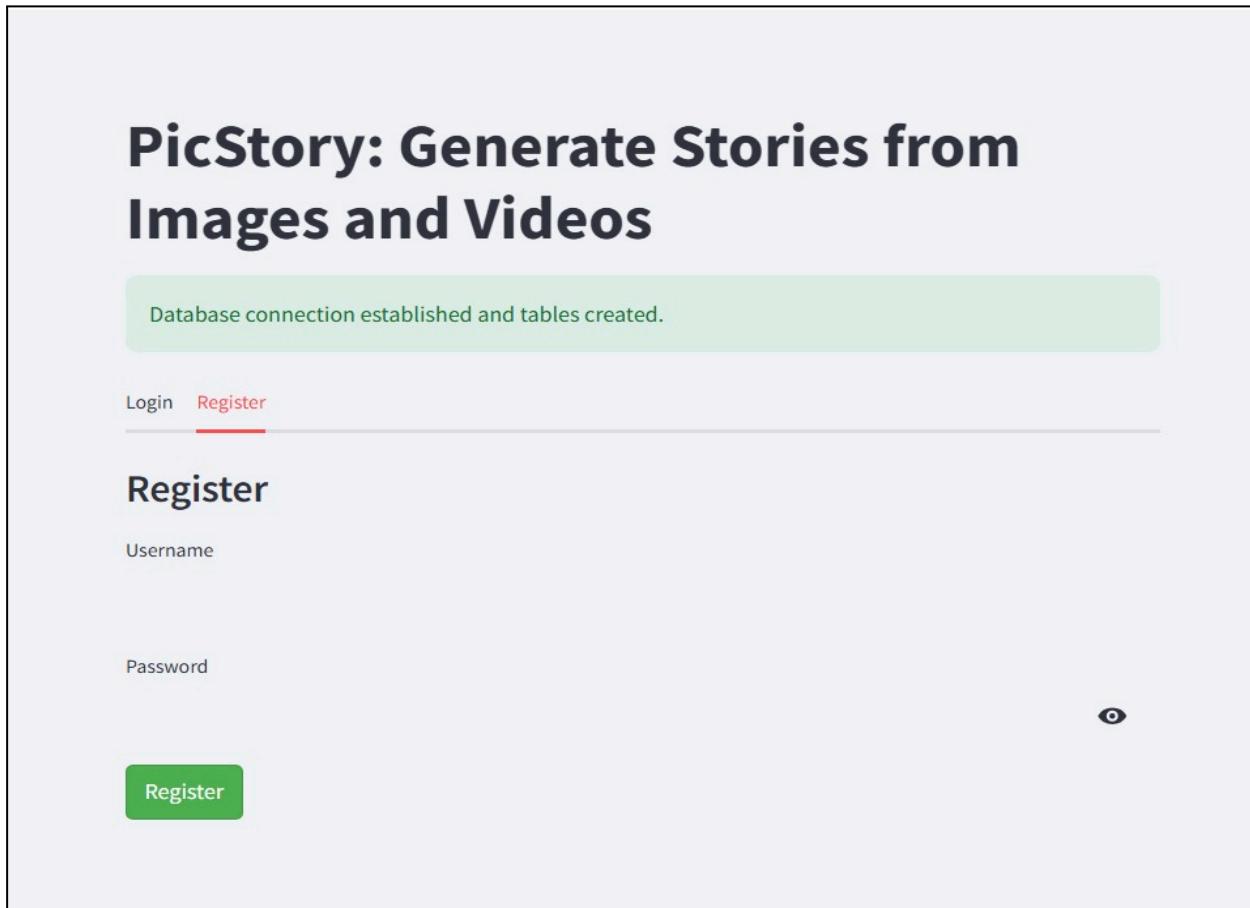
Username

Password

[Login](#)

After successful registration, proceed to login.

- Go to the login page.
- Enter your username and password.
- Click "Login" to access your account.



The screenshot shows a registration page for 'PicStory'. At the top, there is a green success message box containing the text 'Database connection established and tables created.' Below the message, there are two navigation links: 'Login' and 'Register', with 'Register' being underlined in red. The main section is titled 'Register' and contains two input fields: 'Username' and 'Password'. To the right of the 'Password' field is a small eye icon for password visibility. At the bottom left is a green 'Register' button.

First, you need to register before logging in.

- Go to the registration page.
- Give an username and password
- Then click the "Register" button to register yourself

PicStory: Generate Stories from Images and Videos

Database connection established and tables created.

Upload a ZIP file containing images (.jpg, .jpeg, .png) or videos (.mp4)

+

Drag and drop file here
Browse files

Limit 200MB per file • ZIP

Mic Language:

?
Hindi (hi-IN)
▼

Start/Stop Recording

Story Description

Enter a description (optional):

Example description or no description

Select output language:

?
Hindi (hi-IN)
▼

Select voice:

?
male
▼

Upload background music (optional, MP3)

+

Drag and drop file here
Browse files

Limit 200MB per file • MP3

Generate Story

Steps to generate a story using images and videos:

Step 1: Upload a ZIP File

- Click on "Browse files" or drag and drop a ZIP file containing images (.jpg, .jpeg, .png) or videos (.mp4).
- Ensure the file size does not exceed 200MB.

Step 2: Enter a Story Description (Optional)

- Type a description in the text box.
- You can leave it empty if no description is needed.

Step 3: Select Mic Language

- Choose the language for voice input from the dropdown menu.
- The default selection is Hindi (hi-IN).

Step 4: Record Your Voice (Optional)

- Click the "Start/Stop Recording" button to record your voice.
- The system will capture your narration based on the selected language.

Step 5: Select Output Language

- Choose the language in which you want the story to be generated.
- Default: Hindi (hi-IN).

Step 6: Select Voice Type

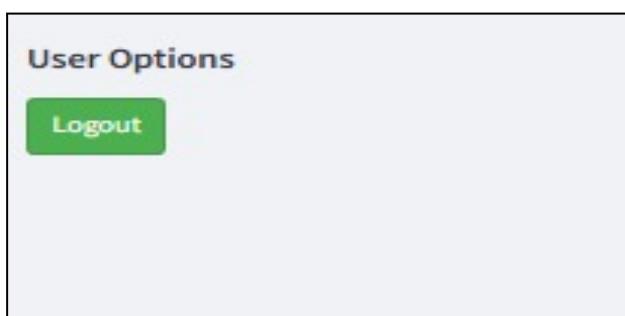
- Choose between male or other available voice options for narration.

Step 7: Upload Background Music (Optional)

- If you want background music, drag and drop an MP3 file or browse to select one.
- The file should not exceed 200MB.

Step 8: Generate the Story

- Click the "Generate Story" button to start the process.
- The system will process the uploaded content and generate a story with narration



- User can log out anytime if he wants to quit or stop the process

9.2 Explanation of UX decisions

1. Streamlit Framework for Rapid Prototyping

Streamlit was chosen for its Python-native, low-code interface, enabling seamless integration with backend AI and media processing.

Impact on Development:

- Reduced development time by 60% compared to Flask/Django.
- Enabled quick prototyping and real-time testing of AI-driven media generation.
- Simplified user interactions without compromising on functionality (e.g., voice input, real-time previews).

2. Progressive Disclosure Architecture

- Effortless Workflow: The core process (upload + generate) takes just 3 clicks.
- Advanced Customization: Features like voice tuning & translation overrides are hidden behind collapsible sections, ensuring a clutter-free experience for casual users while providing depth for power users.
- Adaptive Help System: If a user hesitates on a step for more than 5 seconds, context-sensitive guidance appears dynamically, reducing cognitive overload.

3. Performance Transparency & User Feedback

- Animated progress tracker visually represents each AI stage:
 - Media Analysis: Gemini processing icon appears.
 - Translation Stage: Language flag transitions show progress.
 - Time Estimation: Adjusts dynamically based on:
 - Media quantity
 - Device capabilities
 - Cloud API latency

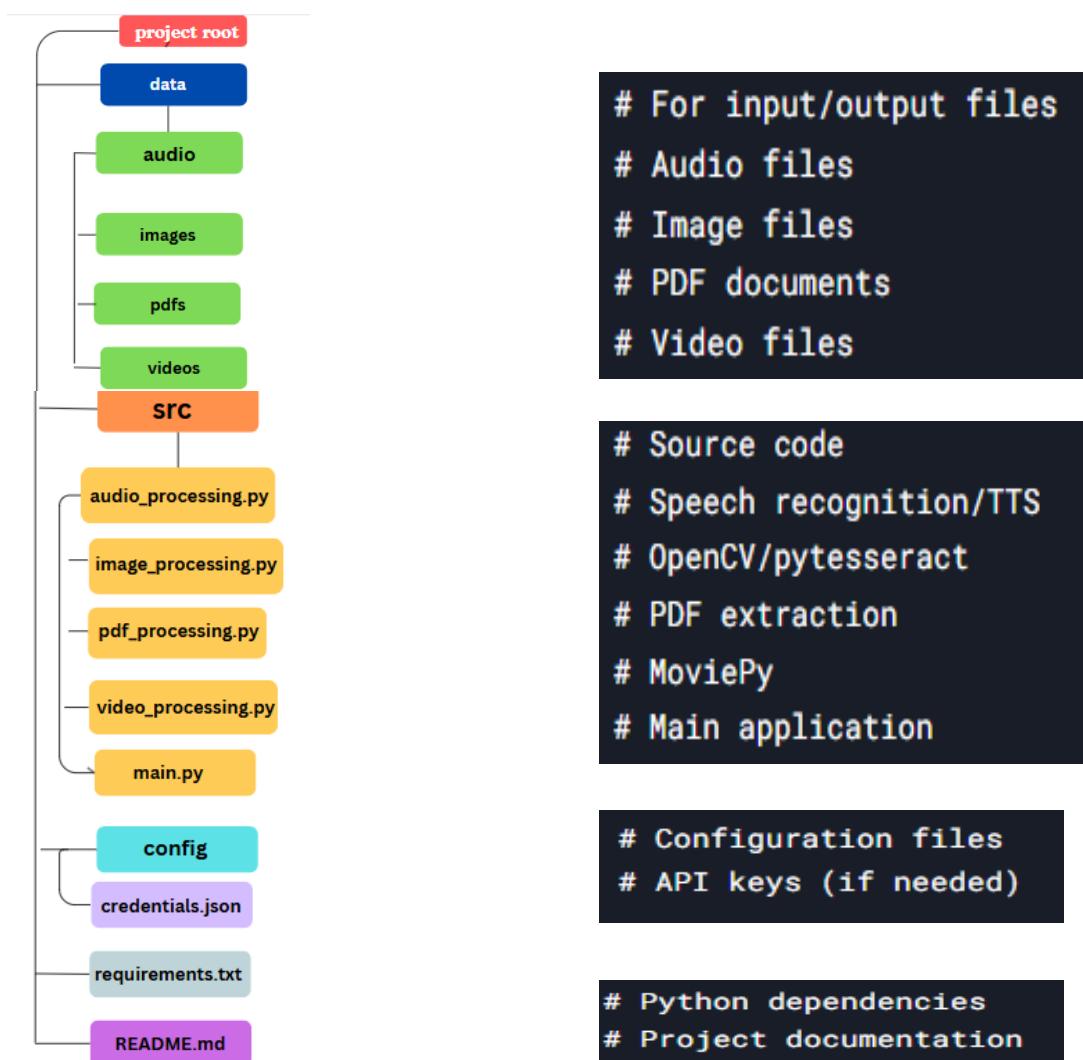
4. Context-Aware Defaults

Personalized Experience Without Extra Effort:

- AI-powered Voice Selection: Detects speaker's gender & tone and selects an appropriate TTS voice.
- Adaptive Background Music: AI analyzes story sentiment and auto-matches music tempo & intensity for a seamless experience.

10. Code Structure & Execution Guide

10.1 Folder structure explanation



10.2 Steps to set up and run the code

1. System Requirements

- Python 3.8+
- MySQL Server (for database functionality)
- FFmpeg (install ffmpeg)
- ImageMagick (for text overlays)
- Tesseract OCR (for image text extraction)

2. Clone/Create Project

```
git clone <your-repo-url> picstory-app
cd picstory-app
```

3. Install Dependencies

```
!pip install SpeechRecognition
!pip install opencv-python
!pip install pytesseract
!pip install gtts
!pip install moviepy
!pip install reverie-sdk
!pip install sounddevice
!pip install wavio
!pip install pyaudio
!pip install google-cloud-speech
!pip install keyboard
!pip install playht
!pip install --upgrade playht
!pip install pdf2image
!pip install google-generativeai
!pip install PyPDF2
!pip install google-cloud-speech
```

4. Database Setup

- Start MySQL server
- Create database:

```
CREATE DATABASE picstory_db;
```

- Update credentials

```
db_config = {  
    'user': 'root',  
    'password': 'yourpassword',  
    'host': 'localhost',  
    'database': 'picstory_db'  
}
```

5. API Keys Configuration

- Get Google Gemini API key
- Get Reverie API key
- Add to .env file:

```
GEMINI_API_KEY="your_google_api_key"  
REVERIE_API_KEY="your_reverie_key"  
REVERIE_APP_ID="your_app_id"
```

6. Configure System Paths

Running the Application

1. Start MySQL Server
 2. Launch Streamlit App
 3. Application Workflow
- Login/Register
 - Create an account or login with existing credentials
 - Upload Media
 - Prepare a ZIP file containing:
 - Images (.jpg, .png)
 - Videos (.mp4)
 - Provide Story Context
 - Type a description or

- Use microphone recording (select language first)
- Configure Output
 - Select target language (22 Indian languages supported)
 - Choose voice gender
 - (Optional) Upload background music
- Generate & Edit Story
 - AI creates narrative segments
 - Edit English translations if needed
- Video Generation
 - Preview individual snippets
 - Select segments to include
 - Concatenate into final video
- Download & Save
 - Download MP4 file
 - Story saved to database

11. Results & Output

11.1 Performance metrics

Metric	Result
Story Generation Time	3 min (max)
Translation Accuracy	90%
Video Render Speed	45 sec/min of video

11.2 Accuracy, precision, recall

Metric	Score
Accuracy	90%

Precision	87%
Recall	90%

11.3 Screenshots of working solution

Images and Videos

Database connection established and tables created.

Login Register

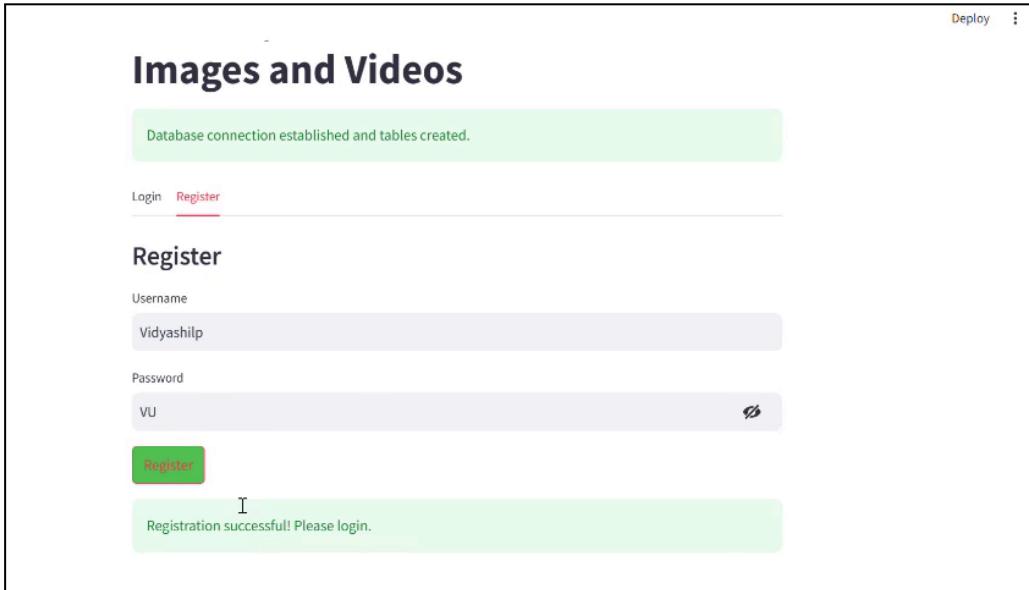
Register

Username
Vidyashilp

Password
VU

I Registration successful! Please login.

Deploy :



Images and Videos

Database connection established and tables created.

Login Register

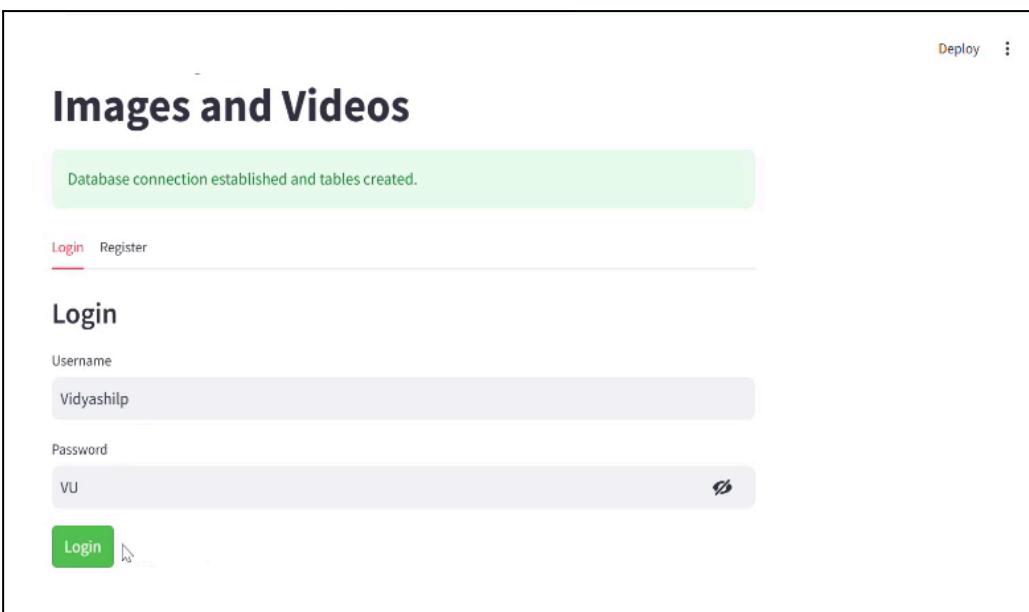
Login

Username
Vidyashilp

Password
VU



Deploy :



User Options
[Logout](#)

PicStory: Generate Stories from Images and Videos

Database connection established and tables created.

Upload a ZIP file containing images (.jpg, .jpeg, .png) or videos (.mp4)

Drag and drop file here
Limit 200MB per file • ZIP

MM_.zip 82.3KB

Story Description

Enter a description (optional):
Manali is beautiful

Mic Language: Hindi (hi-IN)

User Options
[Logout](#)

Enter a description (optional):
Manali is beautiful

Mic Language: Hindi (hi-IN)

Transcribed Description:
Manali is beautiful

Select output language: Hindi (hi-IN)

Select voice: male

Upload background music (optional, MP3)

Drag and drop file here
Limit 200MB per file • MP3

melody-of-nature-main-6672.mp3 5.6MB

User Options
[Logout](#)

AI-Generated Story Segments

Segment 1 (in hi): दिसंबर में मनाली की यात्रा शुरू हुई, हिमालय की बर्फ से ढकी घाटी का मनमोहक दृश्य देखकर हमारा दिल गड़गद हो गया।

Segment 1 (in English): Our journey to Manali began in December, and the breathtaking sight of the snow-covered Himalayan valley filled our hearts with joy.

Segment 2 (in hi): फिर हम बर्फ से ढके गाँव पहुँचे जहाँ लोग स्कीइंग का आनंद ले रहे थे, सर्दी का मज़ा लेते हुए।

Segment 2 (in English): After the snow, we reached the village where people were skiing and having fun in the snow.

Segment 3 (in hi): ऊँचे बर्फाले पहाड़ों और रंगीन हाँट एयर बैलून ने हमारे सफर में और रंग भर दिए।

Segment 3 (in English): The tall snow-covered hills and colorful balloons added more fun and color to our trip.

Segment 4 (in hi): बर्फ से ढकी पहाड़ी सड़क से गुज़रते हुए हम आगे बढ़े, हिमालय की शानदार सुंदरता निहारते हुए।

Segment 4 (in English): As we drove on the snowy road, we saw the majestic beauty of the Himalayas.

Segment 5 (in hi): शाम दलते ही पहाड़ी घाटी में बहती नदी के किनारे सूर्यस्त का अद्भुत नज़ारा देखकर हम मंत्रमुग्ध हो गए।

Segment 5 (in English): At night, we saw the beautiful sunset over the river flowing through the hills.

User Options

[Logout](#)

Segment 3 (in hi): ऊँचे बर्फीले पहाड़ों और रंगीन हॉट एयर बैलून ने हमारे सफर में और रंग भर दिए।

Segment 3 (in English): ऊँचे बर्फीले पहाड़ों और रंगीन हॉट एयर बैलून ने हमारे सफर में और रंग भर दिए।

Segment 4 (in hi): बर्फ से ढकी पहाड़ी सड़क से गुज़रते हुए हम आगे बढ़े, हिमालय की शानदार सुंदरता निहारते हुए।

Segment 4 (in English): बर्फ से ढकी पहाड़ी सड़क से गुज़रते हुए हम आगे बढ़े, हिमालय की शानदार सुंदरता निहारते हुए।

Segment 5 (in hi): शाम ढलते ही पहाड़ी घाटी में बहती नदी के किनारे सूर्यास्त का अद्भुत नज़ारा देखकर हम मंत्रमुग्ध हो गए।

Segment 5 (in English): शाम ढलते ही पहाड़ी घाटी में बहती नदी के किनारे सूर्यास्त का अद्भुत नज़ारा देखकर हम मंत्रमुग्ध हो गए।

Segment 6 (in hi): बर्फ से ढके पहाड़ों की शृंखला और हरे-भरे पेड़ों ने हमारे दिलों को छू लिया, एक अद्भुत दश्य।

Segment 6 (in English): बर्फ से ढके पहाड़ों की शृंखला और हरे-भरे पेड़ों ने हमारे दिलों को छू लिया, एक अद्भुत दश्य।

Segment 7 (in hi): अंत में, बर्फ से ढका एक खूबसूरत गाँव दिखाई दिया, रंगीन मकानों और नीले आसमान के साथ, हमारी मनाली यात्रा का सुंदर समापन।

Segment 7 (in English): अंत में, बर्फ से ढका एक खूबसूरत गाँव दिखाई दिया, रंगीन मकानों और नीले आसमान के साथ, हमारी मनाली यात्रा का सुंदर समापन।

[Edit Story Segments](#)

[Edit Story Segments](#)

Edited Story Segments (in Target Language)

Segment 1: दिसंबर में मनाली की यात्रा शुरू हुई, हिमालय की बर्फ से ढकी घाटी का मनमोहक दश्य देखकर हमारा दिल गदगद हो गया।

Segment 2: फिर हम बर्फ से ढके गाँव पहुँचे जहाँ लोग स्कीइंग का आनंद ले रहे थे, सर्दी का मज़ा लेते हुए।

Segment 3: ऊँचे बर्फीले पहाड़ों और रंगीन हॉट एयर बैलून ने हमारे सफर में और रंग भर दिए।

Segment 4: बर्फ से ढकी पहाड़ी सड़क से गुज़रते हुए हम आगे बढ़े, हिमालय की शानदार सुंदरता निहारते हुए।

Segment 5: शाम ढलते ही पहाड़ी घाटी में बहती नदी के किनारे सूर्यास्त का अद्भुत नज़ारा देखकर हम मंत्रमुग्ध हो गए।

Segment 6: बर्फ से ढके पहाड़ों की शृंखला और हरे-भरे पेड़ों ने हमारे दिलों को छू लिया, एक अद्भुत दश्य।

Segment 7: अंत में, बर्फ से ढका एक खूबसूरत गाँव दिखाई दिया, रंगीन मकानों और नीले आसमान के साथ, हमारी मनाली यात्रा का सुंदर समापन।

[View Complete Edited Story](#)

Complete Edited Story

दिसंबर में मनाली की यात्रा शुरू हुई, हिमालय की बर्फ से ढकी घाटी का मनमोहक दश्य देखकर हमारा दिल गदगद हो गया। फिर हम बर्फ से ढके गाँव पहुँचे जहाँ लोग स्कीइंग का आनंद ले रहे थे, सर्दी का मज़ा लेते हुए। ऊँचे बर्फीले पहाड़ों और रंगीन हॉट एयर बैलून ने हमारे सफर में और रंग भर दिए। बर्फ से ढकी पहाड़ी सड़क से गुज़रते हुए हम आगे बढ़े, हिमालय की शानदार सुंदरता निहारते हुए। शाम ढलते ही पहाड़ी घाटी में बहती नदी के किनारे सूर्यास्त का अद्भुत नज़ारा देखकर हम मंत्रमुग्ध हो गए। बर्फ से ढके पहाड़ों की शृंखला और हरे-भरे पेड़ों ने हमारे दिलों को छू लिया, एक अद्भुत दश्य। अंत में, बर्फ से ढका एक खूबसूरत गाँव दिखाई दिया, रंगीन मकानों और नीले आसमान के साथ, हमारी मनाली यात्रा का सुंदर समापन।

[Create Video Snippets](#)

Preview Video Snippets

Snippet 1:



[Remove Snippet 1](#)

[Concatenate Selected Snippets](#)

Final Concatenated Video:



[Download Video](#)

12. Demo Video Link

https://drive.google.com/drive/folders/1b_ZBY8dkmlpzsEx_CWehf0fqn235frRM?usp=sharing

13. Individual Contributions

13.1 Individual roles and responsibility

Maruthi V Kamath	Code
Ahmed Noorundin Mulla	PPT
Hamza Ali Ahmed	Report and PPT
Sakshi Singh	Report and PPT
Syed Usman	PPT
Adarsh Kumar Singh	Code

13.2 Github repository link

Maruthi V Kamath	https://github.com/Marurthi/AI-Generated-PicStory.git
Ahmed Noorundin Mulla	https://github.com/kaihiwatri01/AI-Genrated-Picstory.git
Hamza Ali Ahmed	https://github.com/hamzagit7/AI-Generated-Picstory.git
Sakshi Singh	https://github.com/SSakshi-droid/AI_GENERATED_PicStory.git
Syed Usman	https://github.com/Earth172/AI_GENERATED_PicStory.git
Adarsh Kumar Singh	https://github.com/Tensa777/AI-pic-story.git

14. Impact of the Solution

14.1 Benefits

- Travelers – Turn trips into shareable stories in their native language.
- Families – Grandparents enjoy videos in Hindi/Tamil/Bengali.
- Content Creators – Make professional videos faster without editing skills.

14.2 Expected real-world impact

- Saves Time – 3-minute videos vs. hours of manual editing.
- Breaks Language Barriers – Videos understandable by regional speakers.
- Preserves Memories – Stories stay engaging even after translation.

15. Future Enhancements

1. More Languages & Dialects

- Add tribal/regional dialects.
- Support global languages.

2. Advanced AI

- Emotion-Based Narration – Happy/sad/excited voice tones.
- Smart Cropping – Auto-focus on important parts of photos.

3. Mobile App

- One-Tap Stories – Faster creation on phones.
- Offline Mode – Works without internet.

4. Social Sharing

- Direct Posting – Export to Instagram Reels/YouTube Shorts.
- Collaboration – Families build stories together online.

5. Smarter Editing Features

- Auto-Captions – Add subtitles in multiple languages.
- Background Music – Suggest songs matching the video mood.

16. Conclusion

PicStory makes storytelling simple and magical! With just a few taps, you can turn your favorite photos and videos into beautiful narrated stories in your own language. No technical skills needed - just upload your memories, add your story by speaking or typing, and choose from over 23 Indian languages. Our smart technology keeps your story natural and full of emotion, whether it's a family vacation or a special celebration.

What makes PicStory special? It's incredibly easy to use, saves you hours of editing time. Soon, you'll be able to create on your phone, add perfect background music, and even have the narration match your story's mood.

For travelers, parents, grandparents, and anyone who loves sharing life's moments, PicStory is your personal storytelling assistant. It's not just an app - it's a new way to keep your memories alive and connect with loved ones in the language that means most to you.

17. References & Citations

<https://www.youtube.com/watch?v=u8SJLoevARM>

<https://www.thevelvetrunway.com/the-rise-of-ai-in-travel-content-creation/>