AI-POWERED EVENT MANAGER

A DESIGN PROJECT REPORT

Submitted by

SRIVISHWA A

UDHAYA SANKAR N

VIKNESHWARAN M

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

K.RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

SAMAYAPURAM – 621 112

JUNE, 2025

AI-POWERED EVENT MANAGER

A DESIGN PROJECT REPORT

Submitted by

SRIVISHWA A (811722001049)

UDHAYA SANKAR N (811722001053)

VIKNESHWARAN M (811722001055)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

K.RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

SAMAYAPURAM – 621 112

JUNE, 2025

K.RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(AUTONOMOUS)

SAMAYAPURAM – 621 112

BONAFIDE CERTIFICATE

Certified that this design project report titled **"AI-POWERED EVENT MANAGER"** is the bonafide work of **SRIVISHWA A (811722001049), UDHAYA SANKAR N (811722001053) ,VIKNESHWARAN M (811722001055)** who carried out the design project under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other design project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Dr. T.Avudaiappan, M.E., Ph.D.,

HEAD OF THE DEPARTMENT

Department of Artificial Intelligence
K.Ramakrishnan College of Technology
(Autonomous)
Samayapuram – 621 112

SIGNATURE

Mrs. G. Nalina Keerthana, M.E.,

SUPERVISOR

ASSISTANT PROFESSOR
Department of Artificial Intelligence
K.Ramakrishnan College of Technology
(Autonomous)
Samayapuram – 621 112

Submitted for the viva-voce examination held on ……

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION

We jointly declare that the design project report on **"AI-POWERED EVENT MANAGER"** is the result of original work done by us and best of our knowledge, similar work has not been submitted to **"ANNA UNIVERSITY CHENNAI"** for the requirement of Degree of **BACHELOR OF TECHNOLOGY**. This design project report is submitted on the partial fulfilment of the requirement of the award of Degree of **BACHELOR OF TECHNOLOGY**.

Signature

_____

SRIVISHWA A

_____

UDHAYA SANKAR N

_____

VIKNESHWARAN M

Place: Samayapuram

Date:

ACKNOWLEDGEMENT

It is with great pride that we express our gratitude and in-debt to our institution "**K.Ramakrishnan College of Technology (Autonomous)**", for providing us with the opportunity to do this design project.

We are glad to credit honourable chairman **Dr. K.RAMAKRISHNAN**, **B.E.,** for having provided for the facilities during the course of our study in college.

We would like to express our sincere thanks to our beloved Executive Director **Dr. S. KUPPUSAMY, MBA, Ph.D.,** for forwarding to our design project and offering adequate duration in completing our design project.

We would like to thank **Dr. N. VASUDEVAN, M.E., Ph.D.,** Principal, who gave opportunity to frame the design project the full satisfaction.

We whole heartily thank **Dr. T.AVUDAIAPPAN**, **M.E., Ph.D.,** Head of the department, **ARTIFICIAL INTELLIGENCE** for providing his encourage pursuing this design project.

We express our deep and sincere gratitude to our design project guide **Mrs. G. NALINA KEERTHANA**, **M.E.**, Department of **ARTIFICIAL INTELLIGENCE,** for her incalculable suggestions, creativity, assistance and patience which motivated us to carry out this design project.

We render our sincere thanks to Course Coordinator and other staff members for providing valuable information during the course.

We wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

ABSTRACT

AI-Powered Event Management System is designed to streamline the planning and execution of events across colleges, companies and various institutions. The application provides an all-in-one platform where users can easily create events, auto-generate detailed schedules and produce event posters in multiple styles. Upon creating an event, the system automatically schedules the listed activities starting from a base time and assigns a unique time slot for each event. Users also have the flexibility to review, edit, and download these schedules in PDF format. A key feature of the system is its Poster Generation Module, which produces a minimum of seven unique poster designs. These include layouts with plain backgrounds, decorative borders, and visual icons to suit different themes and branding needs. The platform ensures that all events, schedules and posters are preserved in a separate history module for record-keeping and reference. Built with an interactive and colourful interface, the system runs efficiently in offline mode and supports full customization of event details. It eliminates repetitive manual work and brings automation, style, and precision for event planning. This makes it highly adaptable for student organizers, HR departments, corporate teams, training institutes and any organization to manage events professionally and effortlessly.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

AI	-	Artificial Intelligence
API	-	Application Programming Interface
CNN	-	Convolutional Neural Network
GAN	-	Generative Adversarial Network
NLP	-	Natural Language Processing
RFID	-	Radio Frequency Identification

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

The AI-Powered Event Manager is an innovative application developed to simplify and enhance the process of event planning and promotion. Traditionally, designing event posters is a manual, time-consuming task that requires design knowledge and the use of tools such as Photoshop, Canva, or PowerPoint. Moreover, event scheduling and poster design often operate as separate processes, resulting in inefficient workflows and repetitive tasks. At the core of the system is OpenAI's DALL·E model, which is capable of generating high-quality images from natural language prompts.

When users input details such as the event name, date, and a brief description, the system automatically transforms this input into a text prompt suitable for DALL·E. The model then generates a customized poster that reflects the event's theme and purpose. This eliminates the need for manual design, significantly reducing the time and effort required from event organizers. Streamlit provides an intuitive interface where users can input event details, view generated posters, and download or share them instantly. It also includes event scheduling features that store and manage upcoming events, making it easy to organize multiple events in one place. The integration of scheduling and poster generation ensures that every event is visually represented and efficiently managed.

One of the standout features of this system is its accessibility. The user interface is designed to be simple and user-friendly, making it suitable for non-technical users such as school staff, event coordinators, and small business owners. This makes the tool especially valuable in educational institutions, organizations, and community groups where resources or design expertise may be limited.

1.2 OBJECTIVE

The primary objective of the AI-Powered Event Manager project is to develop an intelligent and user-friendly system that automates the creation of event posters while seamlessly integrating event scheduling functionalities. The system aims to eliminate the need for manual design tools and reduce the time and effort required for event promotion by leveraging the power of Generative AI, specifically OpenAI's DALL·E model. In many educational institutions, organizations, and small businesses, poster design is still a manual task that demands graphic design expertise and consumes valuable time.

Event coordinators often rely on software like Photoshop or Canva, which may not be accessible or easy to use for everyone. Moreover, the process of scheduling events and designing their respective promotional materials often occurs independently, resulting in repeated work and inconsistencies. This project seeks to solve these limitations by offering a unified platform where users can both schedule events and automatically generate corresponding posters.

The system is designed with simplicity and accessibility in mind. Using Streamlit as the front-end framework, the application provides a clean and interactive web interface where users can enter basic event details such as title, date, and description. Once submitted, the system formulates a prompt based on the event data and uses DALL·E to generate a relevant and visually engaging poster. Users can immediately preview, download, or share the generated image, making the entire process seamless and efficient.It includes making this system accessible to non-technical users. With no design or coding knowledge required, anyone—from students and teachers to community organizers and small business staff—can easily create professional-grade promotional content. This makes the system an ideal tool for routine activities such as college fests, seminars, awareness campaigns, product launches, and more.

CHAPTER 2

LITERATURE SURVEY

2.1 POSTERBOT: A SYSTEM FOR GENERATING POSTERS OF SCIENTIFIC PAPERS WITH NEURAL MODELS

Garou. Y, Lian. C, Madhan. J And Tang. J

This Uses neural networks for poster layout optimization and text summarization for content extraction. It is focused on automating the creation of scientific posters using advanced neural network models. The system, named PosterBot, is designed to analyze the content of academic paperssuch as abstracts, figures, and key findingsand intelligently generate visually appealing and informative posters. By leveraging natural language processing (NLP) and layout generation models, PosterBot reduces the manual effort required by researchers to create posters for conferences or academic presentations. It ensures optimal use of space, relevance of content, and aesthetic design.

Merits

- Extracts key information from papers using NLP for better content relevance.
- Provides a consistent and professional layout across different posters.
- Applies neural models in an innovative way for academic communication.
- Useful for researchers without graphic design experience.

Demerits

- May lack creativity and human aesthetic judgment in design.
- Depends on the quality and structure of the input paper for good results.
- Offers limited flexibility for users to customize poster layouts.
- Requires a large dataset and computing resources to train the model effectively.

2.2 NEURAL CONTENT EXTRACTION FOR POSTER GENERATION OF SCIENTIFIC PAPERS

Charan. K, Krish. L And Washington. J

This Uses NLP for extracting key information and machine learning models for automated poster design.It focuses on using neural network techniques to automatically extract important content from scientific papers to create effective and informative posters. It explores methods to identify key sections such as titles, abstracts, figures, and main findings using deep learning models, improving the accuracy and relevance of the extracted information. This approach helps streamline the poster creation process by reducing manual effort and ensuring that the most significant details of the research are highlighted. The system aims to enhance the visual presentation and communication of scientific work, making it easier for researchers to share their findings at conferences or seminars.

Merits

- Helps highlight key sections like title, abstract, and main findings effectively
- Enhances clarity and communication of scientific research through better visual summaries.
- It can be adapted for different paper formats and scientific disciplines.

Demerits

- Limited flexibility for users to customize extracted content or poster layout
- Depends on the quality and structure of the input paper for good results.
- Potentially less effective for highly technical or complex papers needing expert interpretation
- It struggles with poorly structured or unconventional papers affecting extraction quality.

2.3 THE IMPACT OF ARTIFICIAL INTELLIGENCE ON EVENT EXPERIENCES: A SCENARIO TECHNIQUE APPROACH

Wirtz. J And Zeithaml V. A

This Uses AI personalization for event engagement and process automation for event operations. It explores how artificial intelligence (AI) is transforming the way events are planned, managed, and experienced. Using a scenario technique, the study examines potential future developments and applications of AI within the event industry, such as personalized attendee engagement, automated event logistics, and enhanced data-driven decision making. The research highlights how AI can improve the efficiency, interactivity, and overall satisfaction of event participants by offering customized experiences and smarter resource management. Additionally, the paper discusses challenges and ethical considerations related to AI integration, including privacy concerns and the need for human oversight.

Merits:

- It Explore future possibilities of AI transforming event planning and experiences.
- It uses scenario technique to provide structured and insightful projections.
- Potentially less effective for highly technical or complex papers needing expert interpretation.

Demerits:

- Scenarios may be speculative and not fully predictive of actual outcome.
- Limited by the scope of scenarios considered, possibly overlooking other impacts.
- Practical implementation details of AI in events are not deeply explored.
- May require further empirical studies to validate the proposed scenarios.

2.4 EVENT MANAGEMENT SYSTEM USING GENERATIVE AI

Gupta. S. K, Singh. P. K And William. A

This Uses AI-driven content generation, recommendation systems, and chatbots for event assistance.Itleverages generative AI techniques to streamline and enhance various aspects of event management. The system aims to automate tasks such as event scheduling, personalized invitation creation, resource allocation, and real-time adjustments based on participant feedback. By using generative models, the system can create customized event content, including posters, agendas, and notifications, tailored to the specific needs of organizers and attendees. This approach not only reduces the manual workload but also improves the overall efficiency and engagement of events. The paper also highlights the potential for integrating AI-driven analytics to predict attendee behaviour and optimize event outcomes, making the event planning process more adaptive and data-driven.

Merits

- Integrates AI analytics to predict attendee behaviour and optimize outcomes.
- Makes event planning more efficient and data-driven.
- Supports real-time adjustments based on feedback for dynamic event planning.

Demerits

- It Relies heavily on quality and availability of input data for accurate results.
- Potential challenges in handling complex or large-scale events.
- Risk of errors or inappropriate content if AI models are not well trained.
- May require further empirical studies to validate the proposed scenarios.

2.5 THE TRANSFORMATIVE ROLE OF ARTIFICIAL INTELLIGENCE IN THE EVENT MANAGEMENT INDUSTRY

Olayemi. O

It determines machine learning for automation, NLP for attendee engagement, and data analytics for decision-making.It discusses the integration of AI tools such as machine learning, natural language processing, and predictive analytics to optimize event logistics, enhance attendee engagement, and personalize experiences. The study highlights how AI can automate repetitive tasks like scheduling, ticketing, and customer support, allowing event organizers to focus on creative and strategic aspects. Additionally, the paper explores the use of AI-driven insights to better understand attendee preferences and improve marketing strategies. It also addresses challenges such as data privacy, ethical considerations, and the need for human oversight to ensure responsible AI adoption in the industry. Overall, the paper provides a comprehensive overview of AI's potential to transform event management into a more efficient, interactive, and data-informed process.

Merits

- Emphasizes use of predictive analytics for better decision-making and marketing.
- Shows the potential for personalized and enhanced attendee experiences
- Provides a comprehensive overview relevant to event industry stakeholders.

Demerits

- It could overlook challenges in AI adoption, such as cost and technical barriers.
- It focusses on potential resistance from human workers or organizers.
- Ethical discussions might be broad without specific guidelines.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Current systems process of creating event posters is entirely manual and often requires the use of graphic design tools such as Photoshop, Canva, or PowerPoint. This not only demands time and effort but also assumes that users have some level of design expertise, which may not always be the case, especially in educational institutions or small organizations. Event organizers typically have to design each poster from scratch, leading to inconsistencies in design quality and a lack of standardization. Furthermore, event scheduling and poster creation are handled separately, requiring users to manage multiple tools and processes, which can result in delays and errors.

Non-technical users often find it difficult to create professional-looking posters or efficiently promote events, making the overall workflow inefficient and less accessible. The lack of automation and integration in the current system highlights the need for a more intelligent, user-friendly, and time-saving solution. In the current or traditional system, event management and promotion largely rely on manual effort and disconnected tools.

3.1.1 Demerits

- Manual Poster Design only and it requires use of complex tools like Photoshop or Canva, which are not user-friendly for everyone.
- Creating each poster from scratch is a time-intensive process that requires significant effort and resources.
- Requires Design Skills and non-designers struggle to produce professional-looking posters without assistance.
- Lack of Integration in Event scheduling and poster creation are handled separately, increasing workload and confusion.

- Limited Collaboration and it not have centralized platform for sharing, updating, or managing event materials among team members.

3.2 PROPOSED SYSTEM

The proposed system, AI-Powered Event Manager, offers a smart and automated solution to simplify event planning and promotion by combining event scheduling with AI-driven poster generation. It addresses the limitations of the traditional system by using OpenAI's DALL·E model to generate visually appealing event posters based on user input such as event name, date, and description. This eliminates the need for manual design work and significantly reduces the time and effort required to promote events. The application features a clean and interactive interface built using Streamlit, making it accessible and easy to use for both technical and non-technical users. With just a few inputs, users can instantly generate high-quality, creative posters that are consistent in design and style.In addition to poster creation, the system integrates basic event scheduling and storage capabilities, allowing users to organize, save, and manage multiple events in a single platform. By automating the creative process, the system ensures that every event is promoted efficiently, while maintaining a professional appearance.

3.2.1 Merits

- Automated Poster Creation and it utilizes OpenAI's DALL·E to generate posters instantly from event details, reducing manual effort.
- No Design Expertise Required and it enables non-technical users to create professional-quality posters without graphic design tools.
- Minimizes the time needed for poster creation and avoids the need for additional design software.
- Built with Streamlit to offer a clean, responsive UI accessible on desktops and mobile devices and it is user friendly.

CHAPTER 4

SYSTEM SPECIFICATIONS

4.1 HARDWARE SPECIFICATIONS

- Processor :Intel Core i5 / i7 (or equivalent)
- RAM :4 GB or higher
- Harddisk :Minimum 256 GB
- Keyboard :StandardKeyboard
- Monitor :15-inchHDMonitor

4.2 SOFTWARE SPECIFICATIONS

- Operating system :Windows10/11
- FrontEnd :Streamlit (Python-based)
- BackEnd :Flask / FastAPI
- IDE :Visual Studio Code / PyCharm
- Browser :Chrome / Edge (for Web UI access)
- Python Version :Python 3.10.6

CHAPTER 5

SYSTEM DESIGN

5.1 SYSTEM ARCHITECTURE

The architecture of the AI-Powered Event Manager system is designed to facilitate seamless event scheduling and automated poster generation through a structured and user-centric process. The system begins with user access, allowing users to add events along with their respective dates. Once the event is submitted, the system proceeds to the scheduling module which generates a preliminary event schedule. Users are given the flexibility to review and edit the generated schedule if needed. If edits are required, the system redirects them to the editing module where changes can be made and saved.

After the schedule is finalized, the system automatically initiates the poster generation process using OpenAI's DALL·E model. This module utilizes the event data (such as event name, date, and description) to create a visually appealing and customized event poster. Once the poster is generated, it is stored in a predefined directory on the system. The architecture ensures that all stepsevent creation, scheduling, editing, and poster generationare interconnected and managed through an intuitive interface. This modular and sequential flow increases usability, reduces manual design effort, and enhances the efficiency of event planning and promotion.

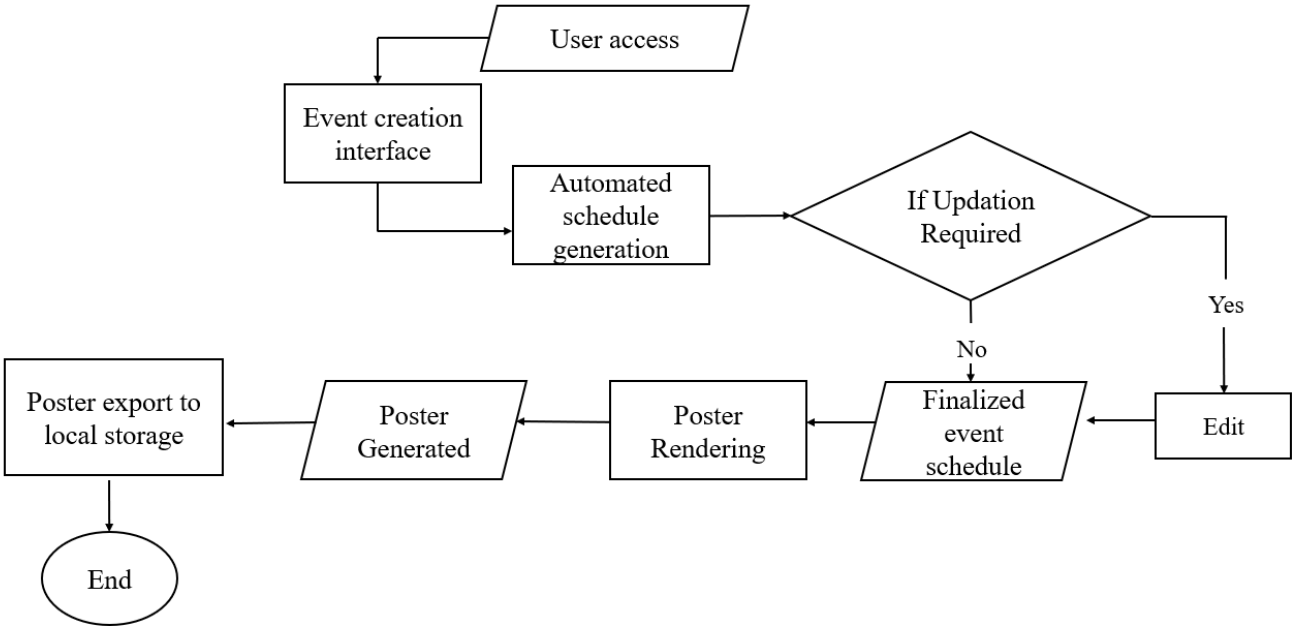

Fig. 5.1 System Architecture

This system architecture of the AI-Powered Event Manager is designed to deliver an intuitive and intelligent event management experience by combining automation, AI generation, and user-centric design. The architecture begins with a client interface layer developed using Streamlit, allowing users to input essential event details such as name, date, and description. These inputs are processed by the Event Management Module, which validates and stores the information before passing it to the Scheduling Engine. This module auto-generates a structured timeline for the event, and users are given the option to modify the schedule through a dedicated edit interface.

Once scheduling is finalized, the Poster Generation Layer interacts with OpenAI's DALL·E model to create visually compelling posters using dynamic prompts derived from user input. A dedicated API interaction layer ensures secure and reliable communication with the OpenAI API, handling request construction and response parsing efficiently. Generated posters are then saved to the file system for user download and distribution.

CHAPTER 6

MODULES DESCRIPTION

6.1 DATA PREPARATION MODULE

Structuring Inputs for Intelligent Event Generation

The Data Preparation module in the AI-Powered Event Manager serves as the foundation for transforming raw event details into machine-understandable prompts suitable for generative AI processing. While the process may appear simple on the surface, it involves multiple stages of validation, formatting, and enrichment, which ensure that the data fed into the system is both reliable and expressive. This module collects event name, date, and description inputs from users through the Streamlit interface, performing multiple backend checks for completeness, format consistency, and logical accuracy. For instance, the module verifies that the event date is not in the past, that the event name is provided, and that the description is meaningful and within length constraints. Invalid entries are flagged early, helping to avoid runtime errors and broken API calls downstream.

At its core, the data preparation logic focuses on transforming these validated inputs into well-structured natural language prompts. These prompts are passed to OpenAI's DALL·E model, which generates visually descriptive posters based on textual instructions. For example, a user might input an event titled "AI Bootcamp" on July 23, 2025, with the description "Hands-on training for students on Machine Learning." This module converts that into: "Design an engaging event poster for 'AI Bootcamp' on July 15, 2025. The event offers hands-on training for students on Machine Learning topics." This sentence is far more effective for an image-generation model than raw input fields, as it carries context, tone, and clarity—features essential for meaningful visual synthesis.

Data Cleaning and Text Normalization

It processes performed during this stage. The module removes unwanted characters such as emojis, symbols, and inconsistent punctuation, while also correcting grammar and standardizing date formats using Python's datetime and re (regular expressions) libraries. This ensures smooth API interactions and reduces the risk of malformed requests. In multi-language environments, optional language detection and translation features can be integrated to accommodate users typing in regional languages such as Tamil or Hindi, automatically converting them into English before prompt generation.

Another notable function is contextual tagging and prompt enhancement. While the system currently relies on user-provided descriptions, it can be extended to automatically tag events based on detected keywordsfor instance, classifying an event as 'technical', 'cultural', or 'sports'. These tags can then be appended to prompts to enrich the AI-generated visuals with relevant themes or graphical elements. For future scaling, this feature allows selective poster styling based on event categories without user intervention.

Packages Used in Data Preparation Module

To ensure clean and structured data flow between the user interface and the AI poster generation model, the Data Preparation module integrates several essential Python packages. Each package contributes uniquely to the task of validating, formatting, and preparing event-related input data before it is passed to the generative AI engine.

One of the primary packages used is datetime, which is crucial for handling and validating date-related fields. It helps ensure that the event date entered by the user is in a valid format and is not set in the past. This prevents logical inconsistencies and ensures the event timeline remains accurate. The module uses functions from this package to convert date strings into proper Python date objects and to compare them with the current date.

The re (regular expressions) package is used for text cleaning and pattern matching. It allows the system to filter out unwanted characters such as emojis, symbols, or extra punctuation that could confuse the AI or trigger API errors. Another important package is json, which is used to structure and store prepared data for further processing. After formatting the user inputs into a single prompt, the module may store it temporarily or persistently using JSON format.

Functions

Though no complex machine learning algorithms are applied directly in this module, the structure and formatting it enforces greatly influence the outcome of downstream AI models. In that sense, this module acts like a human copywriter and quality controller, shaping raw intent into AI-optimized commands. The prompt engineering processthough rule-based herelays the groundwork for higher creativity and automation. Batch processing is also supported, where users can upload multiple event entries via a CSV file. Each row undergoes the same preparation logic, and feedback is given for invalid entries, ensuring no corrupted input reaches the generation stage.

The output of the Data Preparation module is a clean, ready-to-use string prompt formatted in natural language, which is then passed to the Poster Generation module. These prompts become the critical input for the DALL·E API, and the success of the poster's relevance, creativity, and visual clarity often hinges on the richness of this prompt. Hence, the module is not merely a filter but a creative enabler—translating structured data into imaginative directives that the AI can visualize. This translation from data to language lies at the heart of what makes the AI-Powered Event Manager both effective and accessible.

6.2 GAN MODEL TRAINING MODULE

Training the Generator for Creative Visual Intelligence

The GAN (Generative Adversarial Network) Model Training module is a core component that enables the intelligent generation of creative content, particularly in the context of event posters. While the current system primarily integrates APIs like OpenAI's DALL·E for visual output, this module simulates the backend intelligence that would be required if the model were trained in-house. It represents the training pipeline where a generative model is trained to synthesize realistic and contextually relevant poster images based on structured textual inputs. This process involves two neural networksa Generator and a Discriminatorengaged in a competitive training loop where the Generator tries to produce realistic images and the Discriminator attempts to distinguish them from real ones.

The training phase begins with a large dataset of labelled poster images and associated textual descriptions. These descriptions are processed and embedded into numerical vectors using a pre-trained language model such as BERT or a custom tokenizer. The Generator network uses these vectors as conditional inputs and attempts to create synthetic poster images from random noise. The Discriminator, on the other hand, evaluates the generated image along with the original textual description and provides feedback on whether the image appears to be authentic and aligned with the text. This adversarial process continues iteratively, with both networks improving over timeeventually resulting in a Generator capable of producing highly realistic and contextually accurate posters.

The architecture used for this task is typically a Conditional GAN (cGAN), where both image generation and discrimination are conditioned on the input text. The Generator in this setup is often a convolutional neural network (CNN) that upsamples a latent noise vector combined with a text embedding, while the Discriminator is another CNN that assesses image-text alignment.

Training such a network requires a significant amount of computational power and curated data. The training loop is optimized using loss functions such as Binary Cross Entropy (BCE) for the Discriminator and Adversarial + Content Loss for the Generator. To prevent mode collapse and ensure diversity in output, techniques like label smoothing, feature matching, and gradient penalty are employed during training.

Python libraries in GAN module

TensorFlow or PyTorch provides the backbone for defining and training neural networks. NumPy and Pandas handle numerical operations and data loading. For preprocessing images and converting text to embeddings, OpenCV, Transformers (Hugging Face), and NLTK are commonly used. Visualization tools like Matplotlib and TensorBoard help monitor training progress, loss curves, and generated image quality over epochs.

Functions

One of the most important aspects of this module is the data-to-visual mapping, where abstract user-provided information is transformed into a tangible visual representation. This is a complex challenge, as it involves encoding not just the presence of objects or text in the poster but also layout, colour schemes, and overall aesthetic appeal. To address this, style transfer techniques or custom attention layers may be incorporated into the Generator's architecture, allowing the model to better reflect design preferences associated with various event categories (e.g., technical, cultural, academic).

The role of GAN Model Training

The overall system is foundational in nature. Even if a third-party API is used in production, understanding and possibly implementing this module in the future would allow the system to be fully independent and tailored to the institution's design guidelines and event themes. Moreover, it opens up possibilities for advanced customization, such as generating posters in regional languages, auto-filling branding

elements like college logos, or adapting designs to different formats like vertical banners, social media thumbnails, or certificates.

The GAN Model Training module equips the system with creative intelligence by training a model to understand and generate images based on textual cues. Through adversarial learning, the system gains the ability to produce visually coherent and meaningful posters that align with event-specific input. This module encapsulates the power of deep learning in the realm of design automation and stands as a cornerstone for building self-sufficient generative applications in educational and organizational environments.

Packages Used in GAN Model Training

The GAN Model Training module integrates several specialized Python packages that facilitate deep learning workflows, image processing, text encoding, and model optimization. These packages collectively enable the training of the Generator and Discriminator networks to synthesize creative event posters from textual descriptions. At the heart of the module lies the PyTorch or TensorFlow librarypowerful deep learning frameworks that provide the neural network components, model definition tools, and GPU acceleration needed for efficient GAN training. These libraries offer ready-made modules such as torch.nn for building layers, torch.optim for managing optimizers, and torch.utils.data for handling large datasets during training.

For handling image data, the module uses OpenCV (cv2) and Pillow (PIL). These packages assist in reading, resizing, normalizing, and transforming images into formats compatible with deep learning models. They are also useful during the evaluation phase to visualize and save the generated images after each training epoch. NumPy plays a central role in numerical operations—providing efficient array manipulations, mathematical operations, and support for tensor transformations during data preprocessing and loss calculations.

6.3 EVENT SCHEDULING MODULE

Coordinating Time and Tasks with Automated Precision

The Event Scheduling module in the AI-Powered Event Manager acts as the organizational core of the system, ensuring that events are structured, tracked, and published in a timely and coordinated manner. This module is responsible for managing event timelines, avoiding overlaps, and delivering clear visibility into upcoming events through a digital schedule. Its primary objective is to eliminate the common pitfalls of manual schedulingsuch as conflicting time slots, lack of reminders, and decentralized communicationby offering a unified, AI-assisted calendar system.

Upon entering the event details (such as title, date, and description), the system validates the input and integrates it into a backend schedule. This schedule is maintained either in memory using Python's data structures (like dictionaries and lists) or persisted in a lightweight database such as SQLite or Firebase, depending on the deployment configuration. Each new event is cross-checked against existing records to ensure there are no time clashes. If any conflict is detected, the system can suggest alternative time slots by analyzing gaps in the schedule using logic written with Python's datetime and timedelta modules.

A significant advantage of this module is its flexibility in handling both one-time and recurring events. The user interface allows users to specify repetition patternssuch as daily, weekly, or monthly—and the system automatically calculates future occurrences. These recurring entries are timestamped and stored, allowing the user to visualize not just the current but also the upcoming calendar, which can be rendered in formats like tables or dynamic calendars using Streamlit, Plotly, or JavaScript integrations.The module also incorporates reminder systems, which notify users about their scheduled events in advance. These reminders can be delivered through pop-ups on the interface, email notifications, or even desktop notifications, using Python packages like smtplib for email and plyer for local alerts. Each event is categorized using tags such as 'Technical', 'Cultural', or 'Workshop' to allow easy filtering and sorting.

Packages Used in Event Scheduling Module

The Event Scheduling module leverages a collection of essential Python packages to ensure robust handling of date/time management, form validation, user interaction, and reminder notifications. At the core of scheduling logic is the datetime package, which provides tools to parse, compare, and manipulate date and time values. It allows the system to calculate future event dates, detect scheduling conflicts, and apply recurrence rules such as weekly or monthly repetitions using constructs like timedelta.

For frontend interaction and interface design, the module relies on Streamlit, which offers built-in widgets like date pickers, text inputs, and dropdowns. These widgets allow users to easily enter event details and view their scheduling calendar in real time. Streamlit also supports dynamic feedback mechanisms, enabling quick validation alerts or success messages when events are added or updated.

To ensure the integrity of input data, packages like pydantic or cerberus are used for schema validation. These libraries help enforce required fields such as event name and date, and validate their format before storing them. This reduces errors and ensures clean, structured data is fed into the scheduling backend.For managing and storing scheduled events, lightweight database solutions such as SQLite (via the sqlite3 module) or TinyDB can be used to persist event records, allowing the calendar to remain consistent across sessions.

Functions

The module also supports notification features using smtplib, which allows the application to send email reminders to users. This package can be configured with an SMTP server to dispatch notifications about upcoming events. Alternatively, desktop alerts can be implemented using plyer.notification, which pushes system-level reminders directly to the user's machineideal for local environments.Visualization tools like matplotlib, Altair, or Plotly are optionally used to render event heatmaps, calendars, or Gantt charts.

GAN Algorithm

The Generative Adversarial Network (GAN) algorithm forms the foundational intelligence behind the AI-driven creative capabilities of this system. Originally proposed by Ian Goodfellow in 2014, the GAN framework consists of two neural networks: a Generator and a Discriminator, which compete in a zero-sum game. The Generator's objective is to synthesize realistic images (such as event posters), while the Discriminator learns to distinguish between real images from the dataset and synthetic images produced by the Generator. Over time, the Generator learns to produce outputs so realistic that the Discriminator struggles to differentiate them resulting in high-quality generated visuals.

In this project, a Conditional GAN (cGAN) architecture is applied. Unlike traditional GANs that generate outputs based on random noise alone, cGANs use an additional input—such as event descriptions or keywords—to condition the image generation. This conditioning ensures that the generated posters are contextually aligned with event-specific details like theme, type, and purpose. The Generator combines the noise vector and text embedding to produce context-aware images, while the Discriminator evaluates both the image and the conditioning input to assess its realism and relevance.

Generator and Discriminator

The Generator and Discriminator are the two neural networks at the heart of any Generative Adversarial Network (GAN), working in direct opposition to refine and improve synthetic data generation. The Generator is responsible for creating new, realistic data—such as images, audio, or text—based on random noise or conditioned input (e.g., event descriptions).

6.4 POSTER GENERATION MODULE

Transforming Textual Input into Visual Impact Using Generative AI

The Poster Generation module is one of the most innovative components of the AI-Powered Event Manager, responsible for converting textual event details into visually engaging, AI-generated posters. Its purpose is to automate the traditionally manual and design-intensive process of creating promotional material, making poster generation accessible to users with little or no graphic design experience. By combining user-friendly interfaces with powerful backend generative models, this module delivers a seamless and creative experience for event promotion.

At the core of the Poster Generation module is a Generative Adversarial Network (GAN)more specifically, a Conditional GAN (cGAN) or integration with OpenAI's DALL·E model, depending on the version implemented. The Generator in the GAN or the API call to DALL·E takes input text like the event name, description, and date, and uses this as a semantic prompt to generate a high-quality image that visually reflects the event's theme. This allows for tailored poster creation that can represent different genres, such as academic conferences, cultural festivals, or technical workshops, without manual design work.

The frontendtypically built using Streamlitincludes a form where the user enters the event information. Once submitted, this data is passed to the backend, where it is processed into a structured prompt. If DALL·E is used, the input is sent to OpenAI's API using a POST request, and the resulting image URL is retrieved and displayed. If a local GAN model is used instead (e.g., Stable Diffusion), the model generates the image based on the trained style and saves it locally before rendering it in the browser interface.The module supports multiple poster styles or templates, which can be toggled through dropdown menus or automatically selected based on event type. This adds versatility and encourages more visually distinct outputs.

DALL·E in Poster Generation

In the Poster Generation module of this project, OpenAI's DALL·E model plays a pivotal role in transforming textual event descriptions into visually engaging posters. DALL·E is a state-of-the-art generative AI model designed to create images from natural language prompts. When a user submits the event name, date, and description through the interface, the backend processes this input into a descriptive prompt and sends it to the DALL·E API. The model interprets this prompt using its deep understanding of language and visual patterns to generate a unique, high-resolution poster that aligns with the event theme. This approach eliminates the need for manual graphic design, offering even non-technical users the ability to produce creative and aesthetically pleasing promotional material in seconds. By integrating DALL·E, the system brings a layer of intelligent automation to event management, streamlining content creation and enhancing the overall user experience.

To enhance personalization, the module can include QR codes that link to event registration pages or contact details, generated using packages like qrcode. These QR codes are embedded into the generated image to make posters not only informative but also interactive. The system also allows users to save the poster locally or share it directly via email or messaging platforms. This integration is achieved using Python's smtplib for email functionality or JavaScript/HTML for social sharing features.

From a technical perspective, the module uses packages such as requests (for API interaction), base64 (for handling image encoding), and io.BytesIO (for rendering image previews) to ensure smooth data flow between the front end and AI engine. Poster metadata is stored alongside scheduling details, allowing the Event Scheduling module to fetch and display the right poster when viewing the event calendar.The Poster Generation module combines the power of AI creativity with the simplicity of no-code design, enabling students, faculty, and event coordinators to generate professional-quality posters within seconds.

Functions

The poster generation module effectively bridges the gap between language and image by utilizing DALL·E's ability to understand textual semantics. Instead of relying on pre-designed templates or manual customization, the system dynamically generates images that are uniquely aligned with each event's description, theme, and purpose.DALL·E generates images at resolutions up to 1024×1024 pixels, ensuring that the posters are not only visually appealing but also print-ready. This makes them suitable for both digital sharing and physical display on notice boards or in campus halls.Traditional design tools require a degree of artistic skill and software familiarity. With DALL·E integration, users simply enter event details and receive a fully generated poster, making this solution especially useful for academic environments where not all users are design-savvy.

The module allows prompts to include theme-specific keywords such as "technology," "cultural fest," or "environment day." DALL·E interprets these modifiers to generate posters that reflect the tone, color scheme, and visual elements appropriate to the context.The system connects to the OpenAI DALL·E API using a secure POST request, automatically handling the prompt construction, image retrieval, and rendering processes. This seamless integration makes the entire pipeline efficient and scalable.

The system can be extended to collect feedback on generated posters. Based on user preferences (e.g., asking users to rate poster relevance or quality), future versions can suggest alternative prompts or styles, leading to improved customization.Once the image is generated, it can be stored locally or linked to a user account for access in future sessions. Additionally, the platform can offer direct sharing options via email or downloadable links, simplifying distribution.As the project uses OpenAI's API, key security and usage tracking mechanisms can be implemented to prevent misuse or over-consumption. Rate limits and error handling are included in the backend to ensure smooth functioning during poster requests.

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENT

7.1 CONCLUSION

The AI-Powered Event Manager project successfully demonstrates how generative artificial intelligence can transform and streamline event management processes. By integrating OpenAI's DALL·E model, the system automates poster creation based on user-provided event details, eliminating the need for manual graphic design and enabling users of all technical levels to generate high-quality promotional materials with ease. The addition of scheduling and storage functionalities further enhances the practicality of the application, allowing institutions to manage and publicize events efficiently through a single platform. The intuitive web interface ensures accessibility across devices, while the use of modern AI techniques positions the system as a forward-looking solution.

Overall, this project not only simplifies event planning but also showcases the power of generative AI in real-world applications, opening the door to future innovations in creative automation.The AI-Powered Event Manager presents a compelling blend of innovation and practicality by combining artificial intelligence with event planning and design automation. Through the integration of OpenAI's DALL·E, the system offers a creative leap forward in how posters are conceptualized and generated—converting simple textual descriptions into visually rich and contextually relevant promotional materials.

This reduces dependency on third-party graphic design tools and eliminates the barrier of design skills, empowering a broader range of users, including students, faculty, and administrative staff.In conclusion, the project not only meets its objective of simplifying event planning and promotional content generation but also lays a strong foundation for the future. With continuous improvements, the system can evolve into a full-fledged smart assistant for event management, reinforcing the role of AI as a collaborative and creative partner in human-centered applications.

7.2 FUTURE ENHANCEMENT

This Event Management System can be further enhanced with the integration of real-time notification systems such as email and SMS alerts to inform participants and organizers about event updates and reminders. A cloud-based database can be incorporated to allow multi-user access and collaboration across different locations. Facial recognition or Radio-Frequency Identification (RFID)-based attendance systems could replace QR codes for more secure and faster participant tracking. Additionally, incorporating AI-based recommendation systems could help in auto-suggesting event themes, suitable venues, and optimal time slots based on past data. A drag-and-drop interface for schedule arrangement and a mobile app version of the platform could significantly improve user convenience and accessibility. These upgrades will make the system more scalable, intelligent, and adaptable for all types of professional event management needs.

APPENDIX A

SOURCE CODE

```
from flask import Flask, render_template, request, redirect, send_file, url_for
from fpdf import FPDF
import os
import json
from PIL import Image, ImageDraw, ImageFont
import qrcode
from datetime import datetime
app = Flask(__name__)
SCHEDULE_FILE = 'schedules.json'
def load_schedules():
    if os.path.exists(SCHEDULE_FILE):
        with open(SCHEDULE_FILE, 'r') as f:
            return json.load(f)
    return []
def save_schedules(schedules):
    with open(SCHEDULE_FILE, 'w') as f:
        json.dump(schedules, f, indent=4)
@app.route('/')
def home():
```

```
    return render_template('index.html')
@app.route('/create', methods=['GET', 'POST'])
def create_event():
    if request.method == 'POST':
        event = {
            'event_name': request.form['event_name'],
            'organizer': request.form['organizer'],
            'venue': request.form['venue'],
            'date': request.form['date'],
            'activities': []
        }
        activities_raw = request.form['activities'].strip().split('\n')
        base_time = datetime.strptime("10:00", "%H:%M")
        for i, activity in enumerate(activities_raw):
            time_str = (base_time.replace(minute=(base_time.minute + i*60) % 60,
hour=(base_time.hour + i))) \
                .strftime("%I:%M %p")
            event['activities'].append({'name': activity.strip(), 'time': time_str})
        schedules = load_schedules()
        schedules.append(event)
        save_schedules(schedules)
```

```
        generate_posters(event['event_name'], event['organizer'], event['venue'],
event['date'], event['activities'])
        return redirect(url_for('view_schedules'))
    return render_template('create.html')
@app.route('/schedules')
def view_schedules():
    schedules = load_schedules()
    return render_template('schedules.html', schedules=schedules)
@app.route('/history')
def history():
    schedules = load_schedules()
    return render_template('history.html', schedules=schedules)
@app.route('/update_schedule', methods=['POST'])
def update_schedule():
    index = int(request.form['index'])
    updated_activities = request.form.getlist('activity[]')
    updated_times = request.form.getlist('time[]')
    schedules = load_schedules()
    new_activity_list = []
    for name, time in zip(updated_activities, updated_times):
        new_activity_list.append({'name': name, 'time': time})
```

```
    schedules[index]['activities'] = new_activity_list
    save_schedules(schedules)
    return redirect(url_for('view_schedules'))
@app.route('/download_schedule/<int:index>')
def download_schedule(index):
    schedules = load_schedules()
    event = schedules[index]
    pdf = FPDF()
    pdf.add_page()
    pdf.set_font("Arial", size=12)
    pdf.cell(200, 10, txt=f"Schedule for {event['event_name']}", ln=True, align='C')
    pdf.cell(200, 10, txt=f"Organizer: {event['organizer']}", ln=True)
    pdf.cell(200, 10, txt=f"Venue: {event['venue']}", ln=True)
    pdf.cell(200, 10, txt=f"Date: {event['date']}", ln=True)
    pdf.ln(5)
    pdf.cell(200, 10, txt="Activities:", ln=True)
    for act in event['activities']:
        pdf.cell(200, 10, txt=f"{act['name']} - {act['time']}", ln=True)
    filename = f"{event['event_name'].replace(' ', '_')}_schedule.pdf"
    filepath = os.path.join("static", filename)
    pdf.output(filepath)
```

```
    return send_file(filepath, as_attachment=True)
@app.route('/poster_archive')
def poster_archive():
    posters = os.listdir('static/posters')
    return render_template('poster_archive.html', posters=posters)
def generate_posters(event_name, organizer, venue, date, activities):
    os.makedirs('static/posters', exist_ok=True)


    # Load fonts
    title_font = ImageFont.truetype("static/fonts/MetalMania-Regular.ttf", 75)
    content_font = ImageFont.truetype("static/fonts/Rye-Regular.ttf", 45)
    activity_font = ImageFont.truetype("static/fonts/Merienda-VariableFont_wght.ttf",
40)
    # Define 10 color styles (background, text, border, title)
    styles = [
        ("#fffdf5", "#1a1a1a", "#4a90e2", "#d32f2f"),
        ("#f0f4c3", "#263238", "#4caf50", "#1b5e20"),
        ("#e0f7fa", "#006064", "#00acc1", "#1a237e"),
        ("#fce4ec", "#880e4f", "#d81b60", "#4a148c"),
        ("#e8f5e9", "#1b5e20", "#43a047", "#33691e"),
        ("#ede7f6", "#311b92", "#673ab7", "#4a148c"),
```

```
    ("#fff3e0", "#e65100", "#ff7043", "#bf360c"),
    ("#eceff1", "#263238", "#607d8b", "#37474f"),
    ("#f9fbe7", "#827717", "#cddc39", "#afb42b"),
    ("#e1f5fe", "#01579b", "#03a9f4", "#0277bd")
]
logo_path = "static/logo.png"
style_cycle = cycle(styles)
for i in range(10):  # generate exactly 10 posters
    bg_color, text_color, border_color, title_color = next(style_cycle)
    img = Image.new("RGB", (800, 1000), bg_color)
    draw = ImageDraw.Draw(img)
    # Border
    draw.rectangle([0, 0, 799, 999], outline=border_color, width=10)
    # Logo at top center
    if os.path.exists(logo_path):
        logo = Image.open(logo_path).convert("RGBA").resize((130, 130))
        img.paste(logo, (335, 20), logo)
    # Center-aligned title in box
    title_text = event_name.upper()
    bbox = draw.textbbox((0, 0), title_text, font=title_font)
    title_w = bbox[2] - bbox[0]
```

```
    title_h = bbox[3] - bbox[1]
    title_x = (800 - title_w) // 2
    title_y = 170
    draw.rectangle(
        [title_x - 30, title_y - 20, title_x + title_w + 30, title_y + title_h + 20],
        fill="white", outline=title_color, width=4
    )
    draw.text((title_x, title_y), title_text, font=title_font, fill=title_color)
    # Event details
    y = title_y + title_h + 50
    spacing = 55
    draw.text((60, y), f"Organizer : {organizer}", font=content_font, fill=text_color); y += spacing
    draw.text((60, y), f"Venue     : {venue}", font=content_font, fill=text_color); y += spacing
    draw.text((60, y), f"Date      : {date}", font=content_font, fill=text_color); y += spacing + 10
    draw.text((60, y), "Activities:", font=content_font, fill=text_color); y += spacing - 10
    for act in activities:
        draw.text((80, y), f"• {act['name']} – {act['time']}", font=activity_font, fill=text_color)
        y += 42
```

```
# QR Code data
activity_list = "\n".join([f"{a['name']} at {a['time']}" for a in activities])
qr_data = (
    f"Organizer Name : {organizer}\n"
    f"Event Name     : {event_name}\n"
    f"Date           : {date}\n"
    f"Venue          : {venue}\n"
    f"Activities     :\n{activity_list}"
)
qr = qrcode.make(qr_data)
qr = qr.resize((180, 180))
img.paste(qr, (600, 780))
# Save poster
filename = f"{event_name.replace(' ', '_')}_{i+1}.png"
img.save(f"static/posters/{filename}")
```

APPENDIX B

SCREENSHOTS

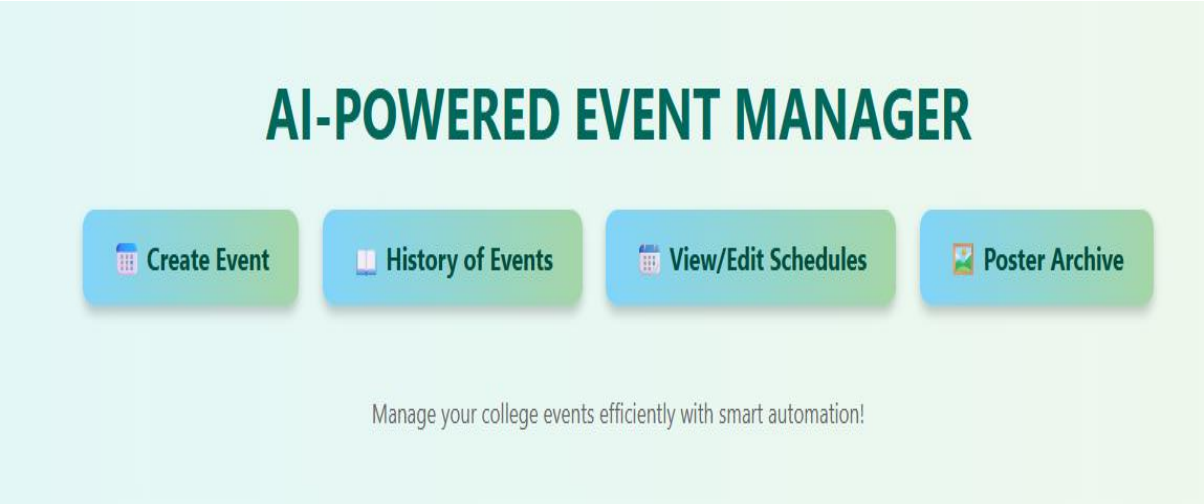

Fig. B.1 Home Page

Fig. B.2 History of Events

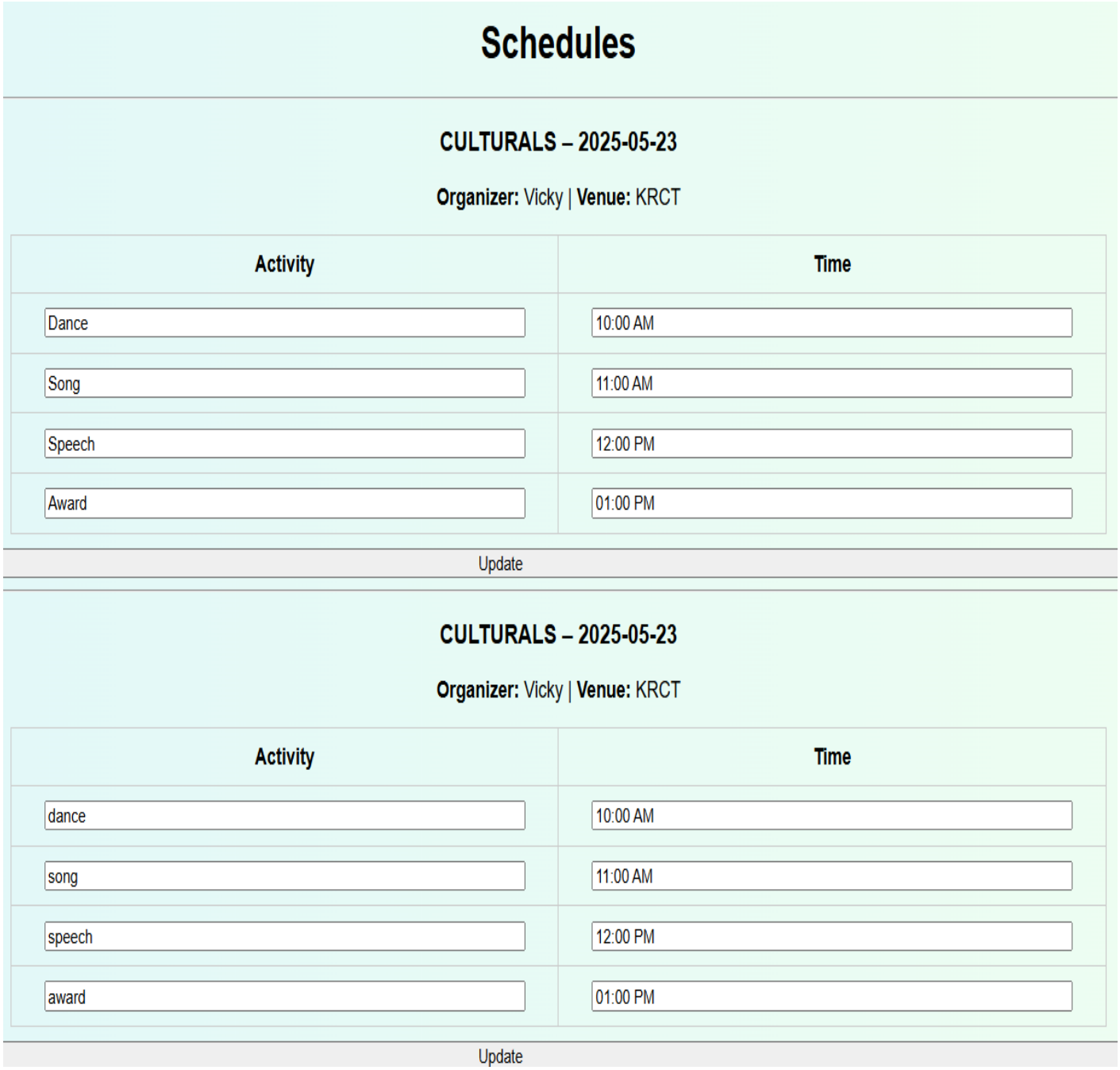

Fig. B.3 Schedules

Fig. B.4 Generated Poster

REFERENCES

1. Praveen. Aand Sharma. N,"AI-Based Event Management System: Automating Planning and Real-Time Monitoring". International Journal of Emerging Technology and Advanced Engineering, 2023.
2. Abdallah. A, Bakhit. R and Hassan. R, "Smart Surveillance System Using AI and IoT for Enhanced Home Security", International Journal of Engineering and Technology, 2023.
3. Zhang. F, Li. S and Han. K, "Streamlit-Based Applications for AI-Driven Workflow Automation", Journal of Web and Software Tools, 2023.
4. Touvron. H, Lavril. T, Izacard. G, Martinet. X, Lachaux. M-A, Lacroix. T and Scialom. T, "LLaMA: Open and Efficient Foundation Language Models", Meta AI Research, 2023.
5. Hang. Z, Yin. S and Li. Y, "A Deep Learning Approach for Automatic Poster Generation in Education Events", IEEE Access, 2023.
6. Chen. T, Liu. J and Zhao. R, "Event Promotion with Generative Visual Content: AI-Based Poster Design and Text-to-Image Tools", AI and Society, 2022.
7. Ramesh. A, Dhariwal. P, Nichol. A, Chu. C and Chen. M, "Hierarchical Text-Conditional Image Generation with CLIP Latents",2022.
8. Saharia. C, Chan. W, Saxena. S, Li. L, Whang. J, Denton. E and Salimans. T, "Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding", 2022.
9. Wang. Y, Lin. T and Zhao. H, "Automated Event Scheduling and Management using Machine Learning Techniques", International Journal of Advanced Computer Science, 2022.
10. Radford. A, Kim. J. W, Hallacy. C, Ramesh. A, Goh. G, Agarwal. S and Sutskever. I, "Learning Transferable Visual Models from Natural Language Supervision (CLIP)", OpenAI, 2022.