

bird- Δ I

Unleash Imagination, Soar with Innovation

Meet THE TEAM

Aditya Kr. Tiwari **22BCE11259**

Ansh Kumar **22BCE11139**

Anurag Kr. Pathak **22BCE10929**

Khushboo Mittal **22BCE11247**

Sivia Anzal **22BCE11350**

Supervised By: Dr. Rajit Nair

Table of C O N T E N T S

01

Abstract

02

Introduction

03

Literature
Review

04

Proposed
Methodology

05

Expected
Outcome

06

Conclusion

07

Reference

ABSTRACT

This abstract involves technical details of our project aimed at constructing a versatile Generative Artificial Intelligence (Gen AI). [1]

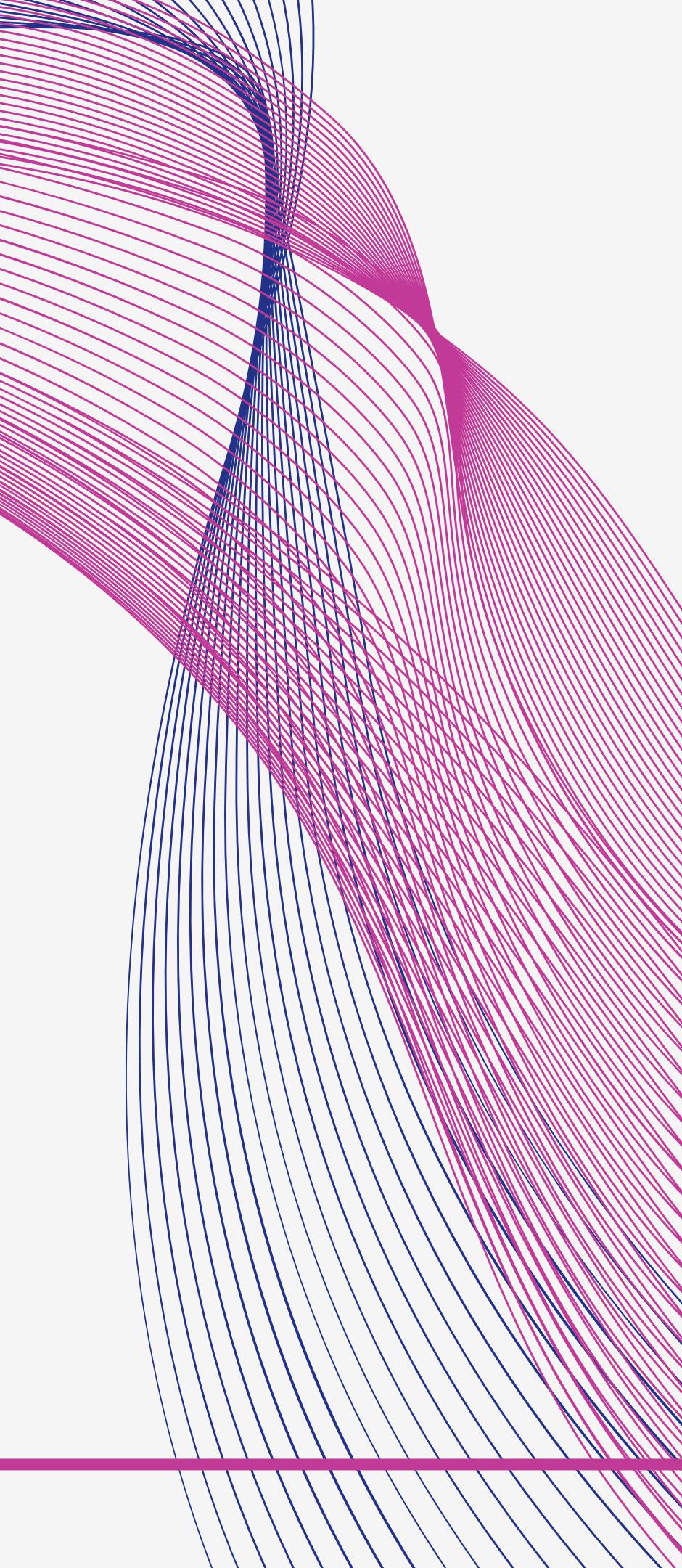
The Primary goal of our project is to develop a Multi- Modal intelligent agent with five foundational features: Image generation, Conversation Initiation, Audio Synthesis, Video Creation and Code Generation, and advanced capabilities capable of autonomous learning, adaptive reasoning, and versatile problem solving.

Incorporating advanced Large language models (LLM) [2] which are trained on deep learning, Machine Learning models and cross-modal interactions, our GPT Model holds a lot of potential.

ABSTRACT

The project's approach to multi-modal AI, include Authentication and API integration. So, integral to the success of our GPT technology exists an authentication system and extensive use of API calls to access external data sources and services. These APIs enable our GPT model to seamlessly integrate with a variety of external platforms, enhancing its versatility and utility across various domains.

The GPT Model is evaluated against strict performance metrics, including a rapid authentication system turnaround time of 300ms, a 250 ms flawless integration with Google login, and a 60ms page loading time. This multidimensional Gen AI prototype represents a substantial leap in diverse AI capabilities, with implications for a wide range of applications across multiple areas.

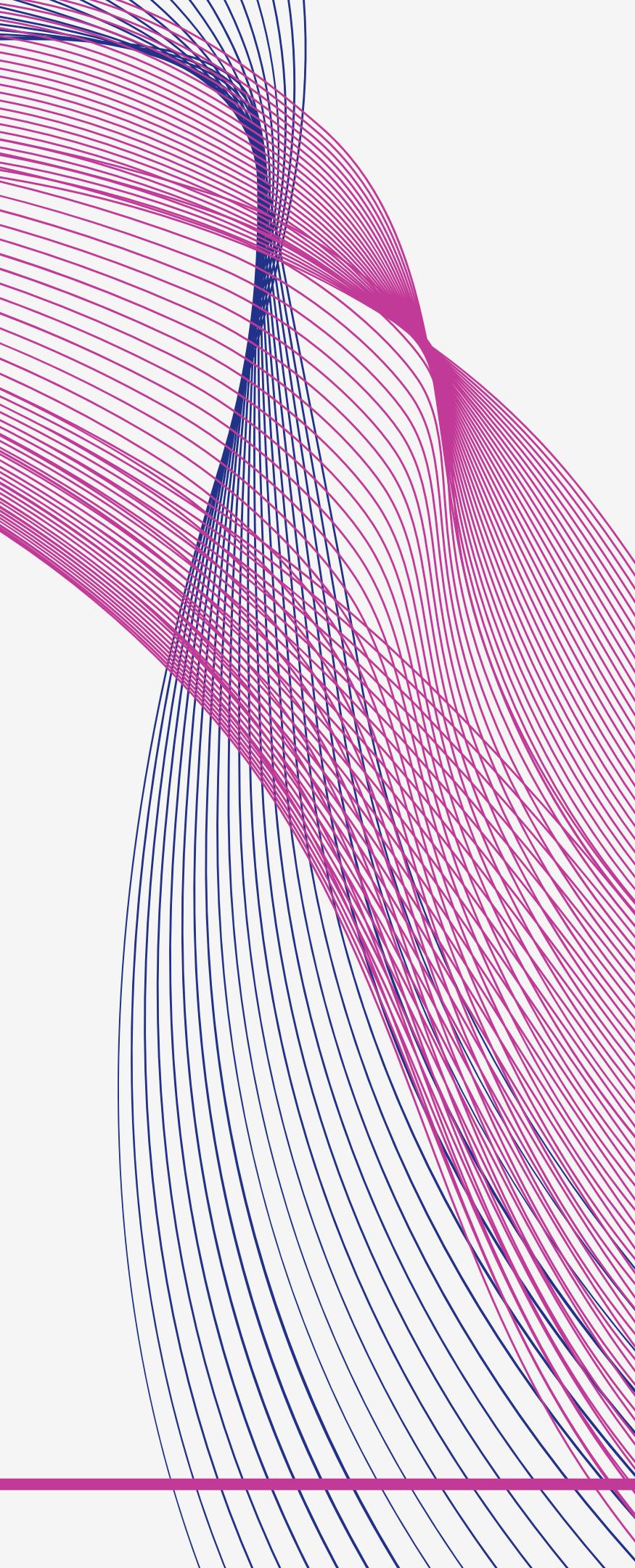


INTRODUCTION

Generative Pre-trained Transformers, are a family of neural network models that uses the transformer architecture and is a key advancement in artificial intelligence (AI) powering generative AI applications. Organizations across industries are using GPT models and generative AI for Q&A bots, text summarization, content generation, and search.

The rise of GPT models is an inflection point in the widespread adoption of ML because the technology can be used now to automate and improve a wide set of tasks ranging from language translation and document summarization to writing blog posts, building websites, designing visuals, making animations, writing code, researching complex topics etc.

In this ever-evolving landscape of artificial intelligence, one landmark achievement stands out as a true game-changer: GPT-3, the third iteration of the Generative Pre-trained Transformer.



The story of GPT began with GPT-1,[3] which pioneered the concept of pre-training a deep neural network on vast amounts of text data. GPT-2[4] emerged as a significant leap forward, boasting an impressive 1.5 billion parameters. It demonstrated the power of large-scale language models and captured the imagination of researchers and enthusiasts alike.

However, it was GPT-3[5] that truly shook the AI landscape to its core. With a staggering 175 billion parameters, GPT-3 is the largest and most capable language model ever created with unparalleled abilities in natural language understanding, generation, and manipulation. Its capacity to perform tasks like translation, text completion, question answering, and even creative writing at human-like levels was nothing short of astounding.

Bird-AI, powered by GPT-3, designed to revolutionize the way we converse, assist, and engage. It represents a pivotal leap forward in AI-driven conversation, offering a transformative tool that holds the promise of enhancing productivity, creativity, and the quality of human-computer interactions like never before.

LITERATURE REVIEW

Closely following the advancements in deep learning and generative Models like Generative Adversarial Networks(GANs)[6] and GPTs, Innovators came up with a lot of technologies that had shown promise in various AI tasks.

Their vision was based on the belief that AI should be capable of creative tasks beyond just language understanding.

Following this vision and collaborating with experts from interdisciplinary segments and combining their knowledge created AI models that could excel in text-image domains and various other domains.

The development of such AI models has been an iterative process involving multiple iterations of training, testing and refining the models.

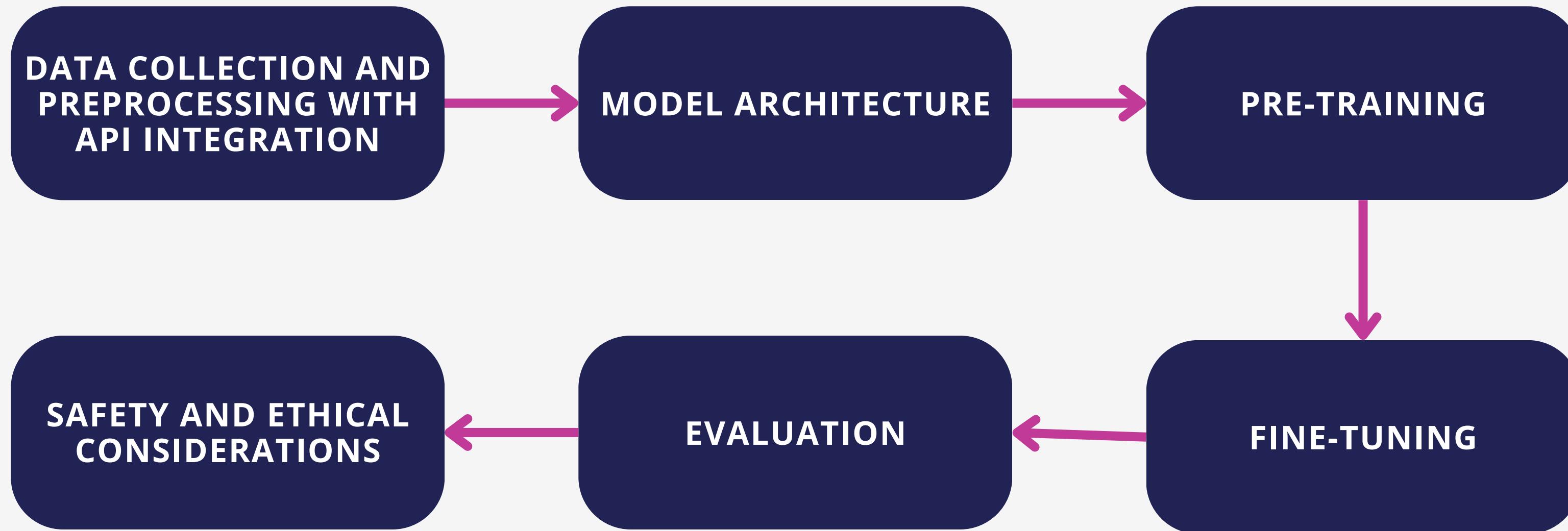
A large portion of building the generative models was dependent on data availability. Collection of large datasets including a wide range was necessary to ensure model's exposure to diverse examples.

LITERATURE REVIEW

The result of these processes resulted in generative AI's like DALL-E [7] , Chat GPT, AI models that could generate high quality texts and images, demonstrating the power of combining the advancements in deep learning with creative thinking and addressing gaps in AI capabilities.

While there are existing generative models on texts and images, they were limited in terms of diversification and in terms of user input lacking in the creation of diversified content which is why recognizing the gap in Multi Modal Intelligent Agent that has diversified fundamental features together, We recognized this specified gap and took away the opportunity to come up with our GPT technology Bird-AI that offers five foundational features: image generation, conversation initiation, audio synthesis, video creation and code generation.

METHODOLOGY



METHODOLOGY

1. Data Collection and Preprocessing with API Integration:

-Data Gathering: Collect a vast and diverse dataset from the internet by integrating APIs. This inclusive dataset should be curated to encompass dialogues from various domains, languages, and contexts to ensure the model's versatility.

- Preprocessing: Clean and preprocess the data to remove noise, anonymize personal information, and format it for training.

2. Model Architecture:

- Transformer Architecture: Choose the Transformer architecture as the foundational model. Transformers have proven highly effective for natural language understanding and generation tasks.

3. Pre-training:

Initiate Pre-train by exposing it to a diverse range of cleaned and preprocessed data. During pre-training, the mode's focus extends beyond language modeling to encompass a broader understanding of multimodal contexts. It learns to predict and generate various aspects of content, including images, audio, text, video, and code.

- Large-Scale Pre-training: Use a vast amount of data and computational resources for pre-training. GPT-3, for example, used 175 billion parameters. More the parameters, more the accuracy.

METHODOLOGY

4. Fine-tuning:

- **Task-Specific Fine-tuning:** Fine-tune the pre-trained model on specific tasks or domains. This involves providing labeled data and customizing the model's output to suit the desired use cases. Fine-tuning may include classification, question answering, or other conversational tasks.

5. Evaluation:

- **Intrinsic Evaluation:** Assess the model's performance on various internal metrics, such as perplexity for language modeling, BLEU scores for translation, or F1 scores for question answering.

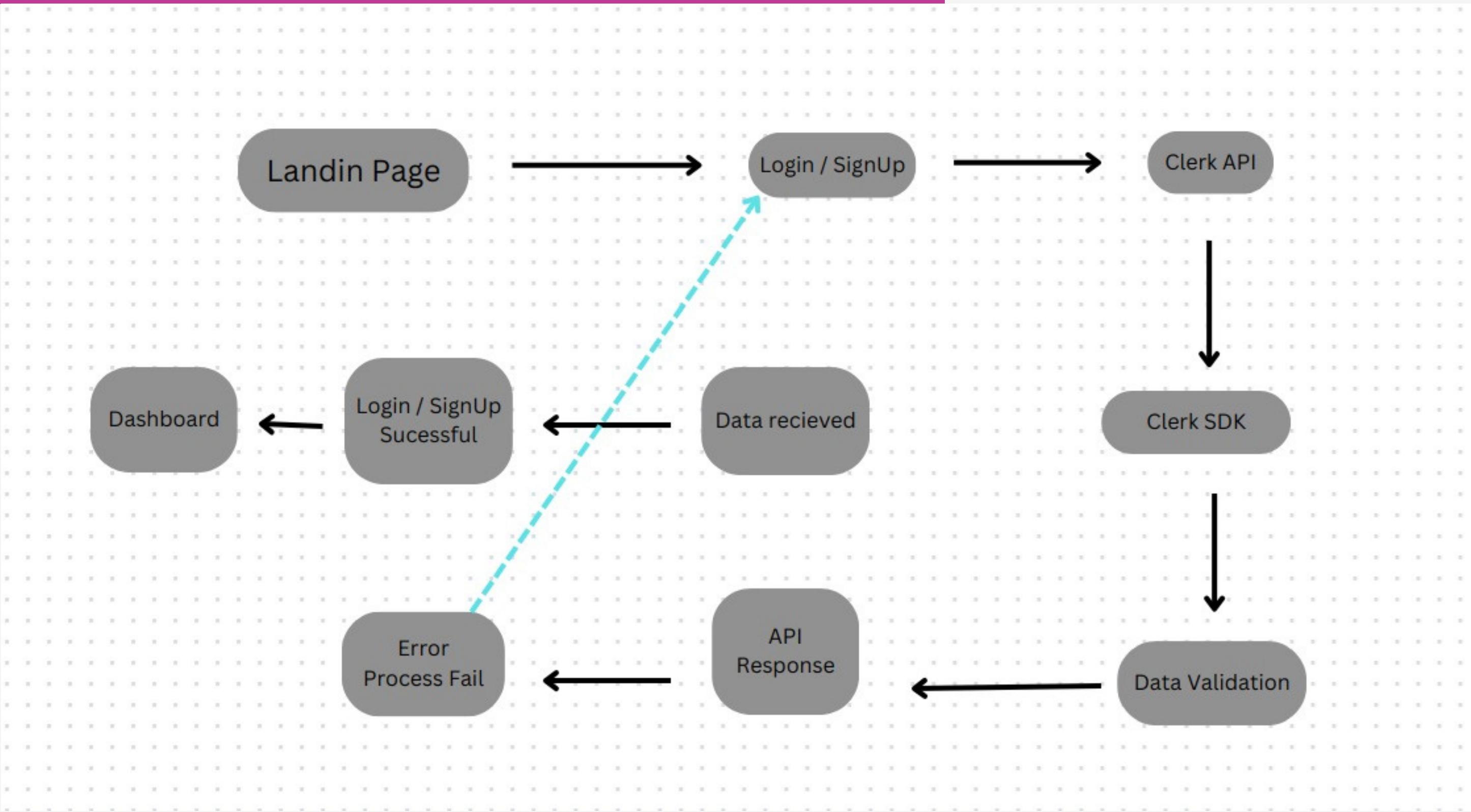
- **Extrinsic Evaluation:** Conduct real-world evaluations to measure the model's performance in practical scenarios. This can involve user studies to assess the quality of generated responses.

6. Safety and Ethical Considerations:

- **Bias Mitigation:** Implement strategies to identify and mitigate biases in the model's responses to ensure fairness and ethical usage.

- **Content Filtering:** Incorporate content filtering mechanisms to prevent the model from generating harmful or inappropriate content.

METHODOLOGY (AUTHENTICATION)



METHODOLOGY (AUTHENTICATION)

1. Install Clerk:

Begin by installing the Clerk SDK or client library in your Next.js app. You can do this by adding the Clerk package as a dependency in your project.

2. Configure Clerk:

Set up the Clerk configuration in your Next.js app. This involves providing the necessary configuration details such as the Clerk API keys or client credentials. These credentials authenticate your app with the Clerk API.

3. User Registration:

Implement the user registration functionality in your Next.js app. You can create a registration form where users enter their details, such as email address and password. When a user submits the form, you can use the Clerk API to register the user and create an account.

4. User Login:

Implement the user login functionality in your Next.js app. Users can enter their login credentials, typically an email address and password, into a login form. Upon submission, you can use the Clerk API to authenticate the user's credentials and log them in.

5. Authentication Tokens: When a user successfully logs in or registers, the Clerk API may provide an authentication token. This token can be used to verify the user's identity across different parts of your application.

METHODOLOGY (AUTHENTICATION)

5. Authentication Tokens:

When a user successfully logs in or registers, the Clerk API may provide an authentication token. This token represents the user's authenticated session. You can store this token securely, such as in a browser cookie or local storage, and include it in subsequent API requests to authenticate the user.

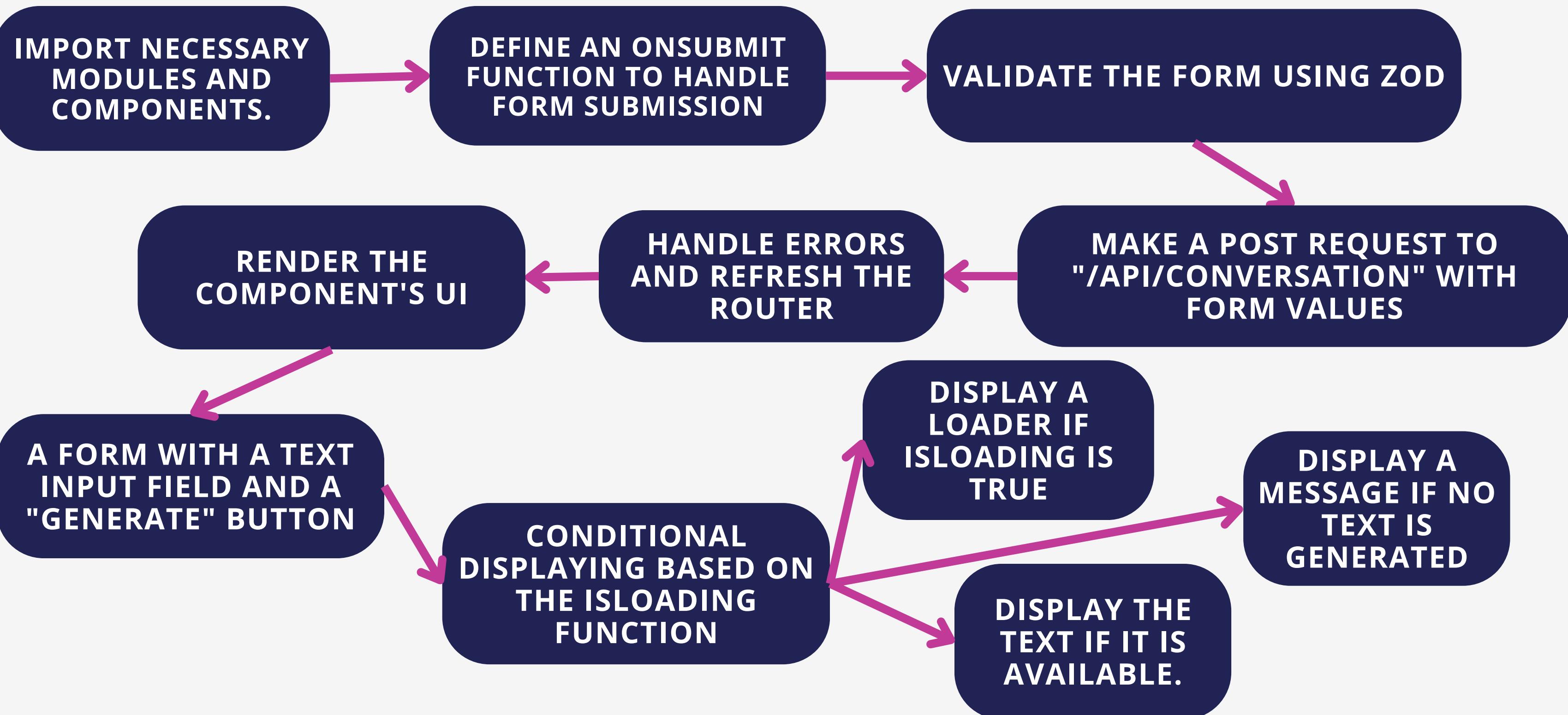
6. Protecting Routes:

Determine which routes or pages in your Next.js app require authentication. You can use the authentication token or other authentication-related information to protect these routes. For example, you can create a higher-order component (HOC) or a custom Next.js middleware that checks for the presence of a valid authentication token before allowing access to protected routes.

7. Handling Authentication Errors:

Handle authentication errors returned by the Clerk API. This could include scenarios such as invalid login credentials, expired authentication tokens, or unauthorized access attempts. You can display appropriate error messages to users and take necessary actions based on the specific error received.

METHODOLOGY (CONVERSATION GENERATION)



METHODOLOGY (CONVERSATION GENERATION)

1. Import Necessary Modules and Components:

In the first step, you gather and include all the required software modules and components to build your video creation tool. This might involve importing libraries and setting up the development environment.

2. Define an onSubmit Function:

You create a function called `onSubmit` to handle what happens when the user submits a form. This function is crucial for processing user inputs.

3. Validate the Form using Zod:

Here, you use the Zod library, which is often used for data validation, to ensure that the information provided in the form is correct and properly formatted.

4. Make a POST Request:

You make an HTTP POST request to a specific endpoint, in this case, "/api/Conversation," to send the data collected from the form to your chatbot or AI model for generating text.

METHODOLOGY (CONVERSATION GENERATION)

5. Handle Errors and Refresh the Router:

This step involves managing any errors that might occur during the process, and if necessary, refreshing the routing or navigation within your application to ensure a smooth user experience.

6. Render the Component's UI:

You design and display the user interface of your video creation tool, which typically includes input fields for text and a button for generating the video.

7. Create a Form:

This form includes at least one text input field where users can provide the input for generating the video and a "Generate" button to initiate the process.

8. Conditional Displaying Based on isLoading Function:

Depending on the status of the `isLoading` variable, you manage what the user sees:

METHODOLOGY (CONVERSATION GENERATION)

- 8a. Display a Loader:

If `isLoading` is true, you show a loading indicator (such as a spinner) to inform the user that the system is processing their request.

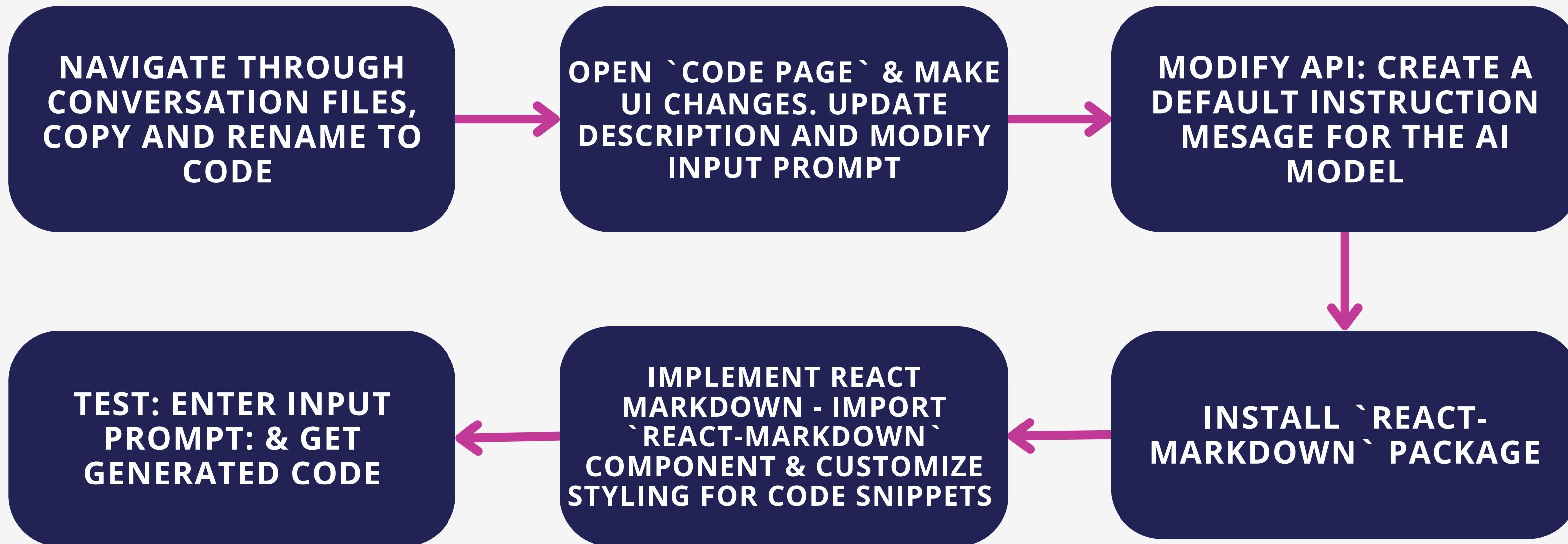
- 8b. Display a Message:

If the generation process doesn't yield any text, you display a message to the user to indicate that no content was generated.

- 8c. Display the Generated Text:

If the text is successfully generated, you display it to the user so they can see the result of the AI chatbot's work.

METHODOLOGY (CODE GENERATION)



METHODOLOGY (CODE GENERATION)

1. Navigation through folders:

- Navigate through app, dashboard, and routes folder
- Drop Copy from the conversation folder

2. Code Page:

- Make UI changes
- Update description
- Modify input prompt “Write simple toggle button using React Hooks”

3. Modify API:

- Create a default instruction message for the AI Model:

Role: System

Content: “You are a code generator. Answer only in markdown snippets. Use comments for explanations.”

- Modify response to include the instruction message before other messages
- Add an error message for code-related errors.

METHODOLOGY (CODE GENERATION)

4. Install React Markdown:

- Install `react-markdown` package.

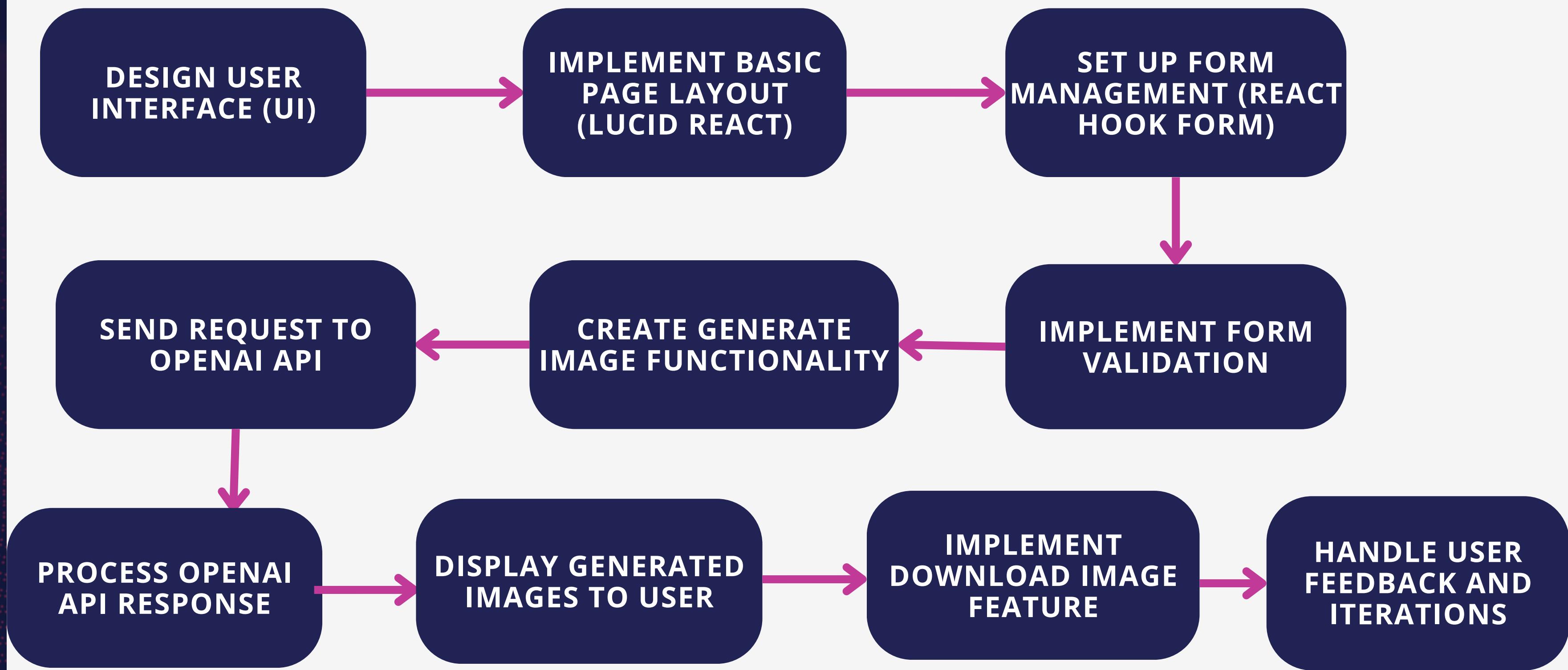
5. Implement React Markdown

- Import react-markdown component
- Replace message content rendering with `react-markdown` component
- Customise styling for code snippets

6. Generate Code

- Input prompt: “Model using Zod and Tailwind”
- Get generated code
- Render code in markdown format
- Apply styling for snippets

METHODOLOGY (IMAGE GENERATION)



METHODOLOGY (IMAGE GENERATION)

1. Design User Interface (UI):

- Plan the user interface layout and design, considering the input fields, buttons, and overall aesthetics.

2. Implement Basic Page Layout (Lucid React):

- Create the initial layout of the image generation component using Lucid React, incorporating components for user input and image display such as heading, description, icon, icon colour, bg colour etc.

3. Set Up Form Management (React Hook Form):

- Integrate React Hook Form for efficient management of form data, including text-based user inputs for image generation, selecting amount of images and resolution of images.

4. Implement Form Validation:

- Add form validation logic to ensure that user inputs are valid and meet the required criteria.

5. Create Generate Image Functionality:

- Develop the functionality to trigger image generation when the "Generate Image" button is clicked.

METHODOLOGY (IMAGE GENERATION)

6. Send Request to OpenAI API:

- Implement the API configuration code to send an API request to the OpenAI server, including the user's text description and parameters.

7. Process OpenAI API Response:

- Handle the response from the OpenAI API, extract image URLs, and process the generated images.

8. Display Generated Images to User:

- Integrate components and extract image URLs to display the generated images within the chatbot's user interface.

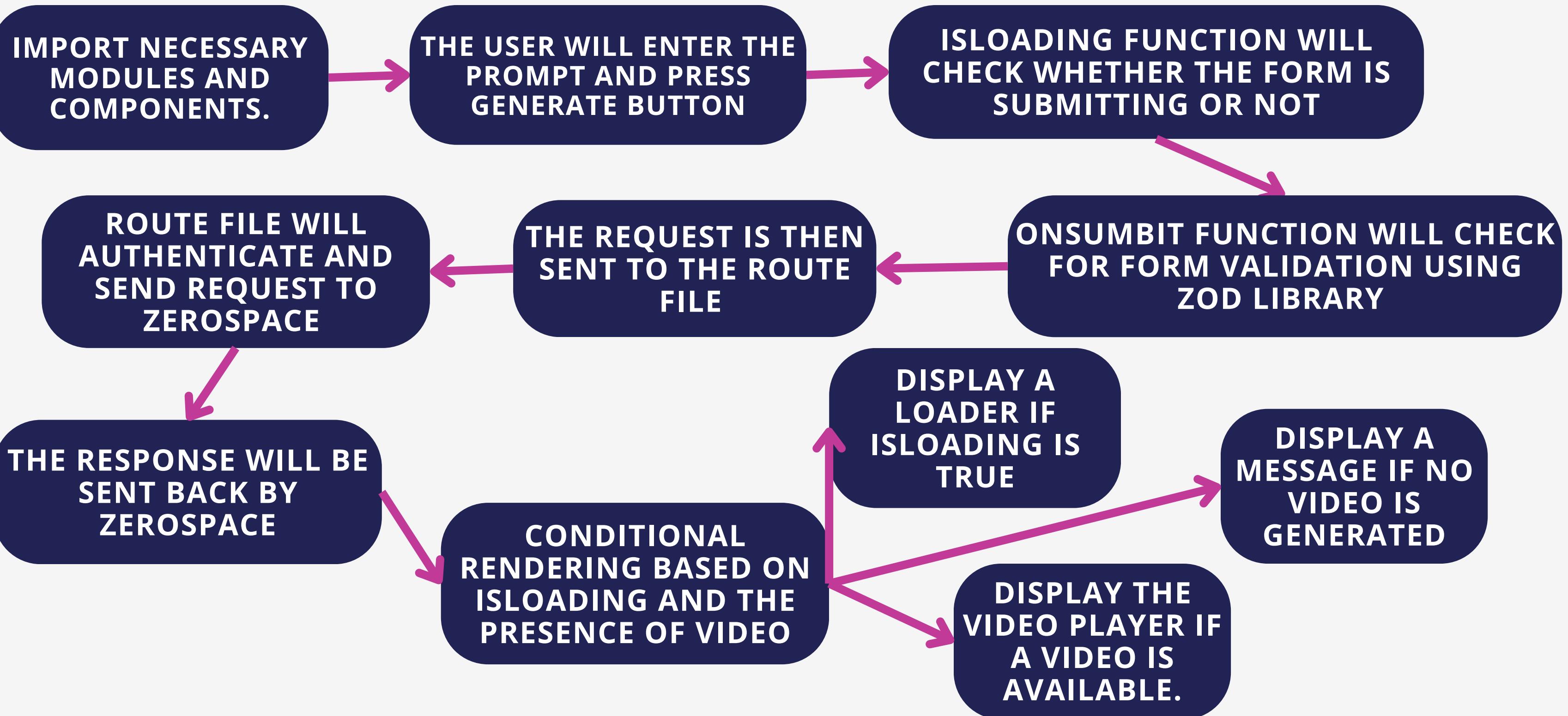
9. Implement Download Image Feature:

- Add the functionality for users to download the selected generated image.

10. Handle User Feedback and Iterations:

- Implement mechanisms for users to provide feedback and iterate on image generation requests as needed.

METHODOLOGY (VIDEO GENERATION)



METHODOLOGY (VIDEO GENERATION)

1. Import necessary modules and components:

In the first step, the required software modules and components, such as libraries and tools, are imported into the chatbot's code. These components are essential for the chatbot to function properly.

2. User input and generation:

The user interacts with the chatbot by entering a prompt, which is a text-based instruction or request. After providing the prompt, the user initiates the video generation process by clicking a "Generate" button.

3. isLoading function:

This function is responsible for checking whether the form is currently being submitted. It helps manage the user interface's state, indicating whether the system is actively processing the request.

4. onSubmit function:

The `onSubmit` function is responsible for validating the user's input using the Zod library. It ensures that the user's prompt adheres to specific criteria or rules, and if the input is valid, it proceeds to the next steps.

METHODOLOGY (VIDEO GENERATION)

5. Sending the request:

Once the user's input is validated, the chatbot sends a request to the route file. This request contains the user's prompt and any necessary information for generating the video.

6. Route file and authentication:

The route file is responsible for handling incoming requests, including authentication, if required. It ensures that only authorized requests are forwarded to the next step, which is Zerospace.

7. Zerospace response:

Zerospace is a component that generates the video content based on the user's prompt. It processes the request and sends back a response, which is the generated video.

8. Conditional rendering:

This step deals with displaying the appropriate content on the user interface based on various conditions:

.

METHODOLOGY (VIDEO GENERATION)

8(a) Loader:

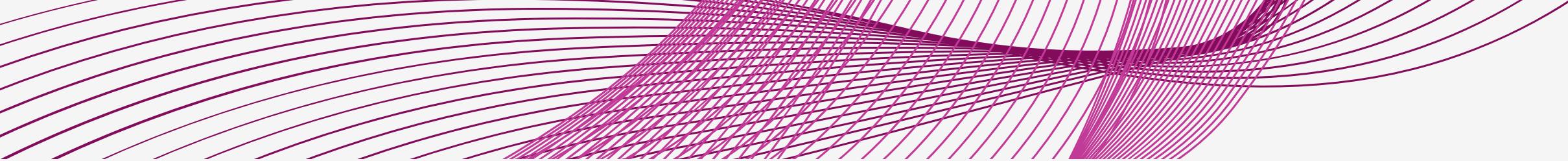
If the system is still processing the request (`isLoading` is true), a loading animation or message is displayed to indicate to the user that the video is being generated.

8(b) Message:

If no video is generated or if there is an error in the process, a message is displayed to inform the user of the issue.

8(c) Video player:

If the video is successfully generated, a video player is displayed on the user interface, allowing the user to view and interact with the generated content.



OUTCOMES

The expected outcomes of the Bird AI project are centered around the development of a cutting-edge Generative Artificial Intelligence (Gen AI) system with a wide range of capabilities and seamless integration into various applications and platforms. Key anticipated outcomes include:

Multi-Modal Intelligence

01

Bird AI is expected to excel in various modes of interaction, including image generation, audio synthesis, video creation, conversation initiation, and code generation. This versatility will make it a powerful tool for a diverse set of tasks.

Advanced Learning and Reasoning

02

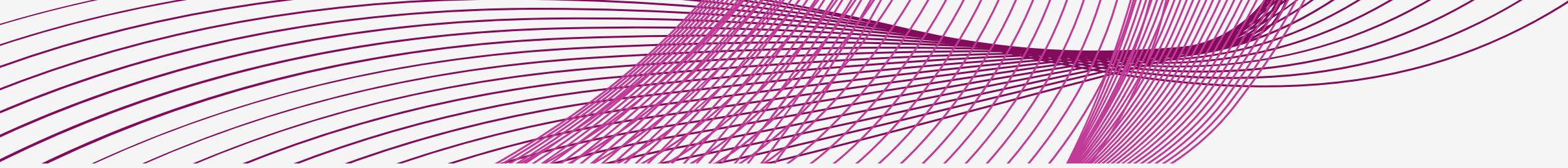
The project aims to imbue Bird AI with autonomous learning capabilities, adaptive reasoning, and versatile problem-solving skills. This means the AI can continually improve its performance and adapt to new challenges.

Integration of Large Language Models

03

By leveraging Large Language Models (LLM) trained through deep learning and machine learning techniques, Bird AI is expected to have a strong foundation in natural language understanding and generation.





OUTCOMES

04

Authentication System

The inclusion of an authentication system ensures secure and authorized access to Bird AI's capabilities, which is crucial for protecting sensitive data and maintaining privacy.

05

API Integration

Extensive use of API calls for accessing external data sources and services is a fundamental outcome. This integration enhances Bird AI's adaptability, allowing it to interact with a wide range of external platforms and data, making it more versatile and useful across various domains.

06

Cross-Modal Interactions

The project's focus on cross-modal interactions means that Bird AI can seamlessly combine different types of data and sensory inputs, making it capable of handling complex, real-world tasks that involve multiple modalities.

07

Enhanced Versatility

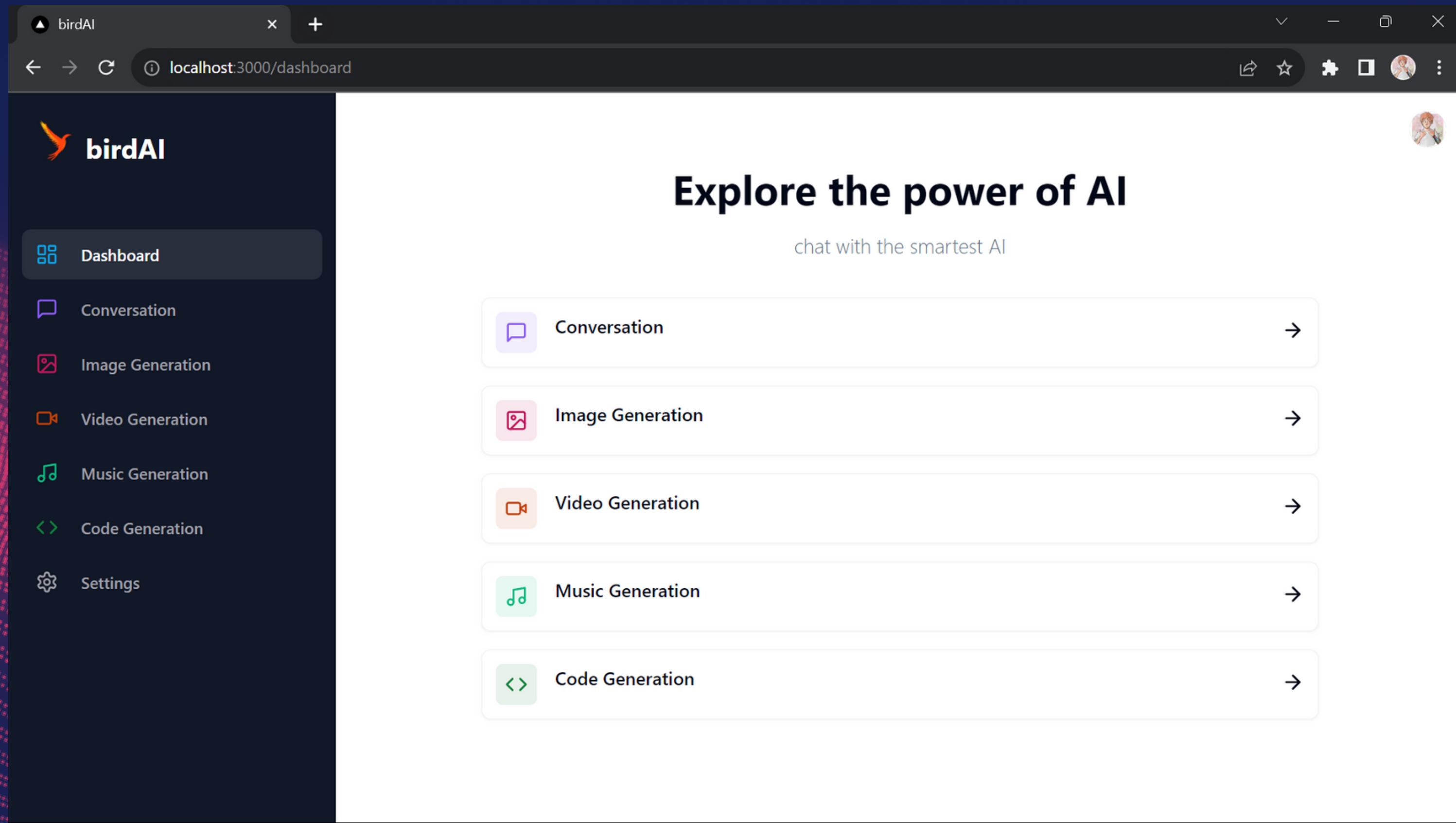
Overall, the expected outcome is an AI system that is highly versatile and capable of performing tasks ranging from creative content generation to solving complex problems, making it a valuable asset in a wide array of industries and applications.

OUTCOME

The image displays four screenshots of the birdAI application, illustrating its landing page, sign-in process, dashboard, and main features.

- Landing Page:** Shows a simple interface with a "Landing Page" header, a "login" button, and a "Register" button.
- Sign in:** A modal window titled "Sign in" asking "to continue to Bird-AI". It offers two options: "Continue with Google" and "Email address".
- Dashboard:** The user's profile information (Anurag Pathak, anuragpathakoff@gmail.com) is shown, along with "Manage account" and "Sign out" buttons. The sidebar includes links for Dashboard, Conversation, Image Generation, Video Generation, Music Generation, Code Generation, and Settings.
- Main Features:** A central panel titled "Explore the power of AI" with the subtext "chat with the smartest AI". It lists six features with corresponding icons: Conversation, Image Generation, Video Generation, Music Generation, Code Generation, and Settings.

OUTCOME



OUTCOME

The image shows a screenshot of the birdAI web application. On the left, there is a dark sidebar with the birdAI logo at the top. Below it is a navigation menu with the following items: Dashboard (blue icon), Conversation (purple icon, currently selected), Image Generation (camera icon), Video Generation (video camera icon), Music Generation (musical note icon), Code Generation (code icon), and Settings (cog icon). The main content area has a light background. At the top, it says "Conversation Our AI ChatBot". Below that is a text input field containing "what can i tell?" and a "Generate" button. In the center, there is a circular illustration of a person looking through binoculars at a globe. At the bottom of the main content area, it says "No conversation started.". To the right of the main content area, there is a vertical sidebar with the same navigation menu as the sidebar on the left. The "Conversation" item in this sidebar is also highlighted with a purple background. The overall theme is a clean, modern design with a focus on AI and generative capabilities.

birdAI

Conversation
Our AI ChatBot

what can i tell?

Generate

No conversation started.

birdAI

Conversation
Our AI ChatBot

what can i tell?

Generate

Artificial intelligence (AI) refers to the simulation of human intelligence in machines that are programmed to think, learn, and problem-solve like a human being. It involves the development of algorithms and technologies that enable computers to perceive and comprehend the environment, reason and make decisions, and adapt and learn from experience. AI can be further categorized into narrow AI, which is designed for specific tasks, and general AI, which aims to possess human-like intelligence across various domains. Some common applications of AI include speech recognition, natural language processing, computer vision, robotics, and virtual assistants.

what is artificial intelligence?

OUTCOME

The screenshot shows the birdAI web application interface. The title bar says "birdAI" and the URL is "localhost:3000/code". The sidebar on the left lists various AI generation services: Dashboard, Conversation, Image Generation, Video Generation, Music Generation, and Code Generation (which is selected). Below the sidebar is a navigation menu with "Code Generation" and "Settings". The main content area features a "Code Generation" header with the sub-instruction "Generate code using descriptive text". A text input field contains the query "ask me to write a code if you can't find it on stackoverflow". To the right of the input is a "Generate" button. Below the input field is a cartoon illustration of a person looking through binoculars at a globe. At the bottom of the main area, the message "No conversation started." is displayed.

The screenshot shows the generated code snippet. It starts with a comment: "Sure! Here's an example code snippet in Python to find prime numbers:". The code is as follows:

```
def is_prime(n):
    if n <= 1:
        return False
    if n == 2:
        return True
    if n % 2 == 0:
        return False
    i = 3
    while i * i <= n:
        if n % i == 0:
            return False
        i += 2
    return True

def get_prime_numbers(limit):
    prime_numbers = []
    for num in range(2, limit + 1):
        if is_prime(num):
            prime_numbers.append(num)
    return prime_numbers

# Example usage
limit = 100
print(get_prime_numbers(limit))
```

Below the code, a note states: "This code includes two functions: `is_prime(n)`: This function checks whether a given number `n` is prime or not. It uses the trial division method, where you iterate from `2` to the square root of `n` to check if `n` is divisible by any number in between. `get_prime_numbers(limit)`: This function generates a list of prime numbers up to a given `limit` number. It iterates from `2` to `limit` and uses the `is_prime()` function to check if each number is prime or not. In this example, calling `get_prime_numbers(100)` will return a list of prime numbers from `2` to `100`."

At the bottom of the page, there is a text input field with the placeholder "write a code to find prime numbers".

OUTCOME

The screenshot displays the birdAI web application interface, specifically the Image Generation section. On the left, a dark sidebar menu lists various AI services: Dashboard, Conversation, Image Generation (which is selected and highlighted in pink), Video Generation, Music Generation, Code Generation, and Settings. The main content area features a header "Image Generation" with the sub-instruction "Turn your prompt into an image". Below this is a text input field containing the prompt "A picture of an amphibious yordle holding a trident", followed by a "Generate" button. A small, semi-transparent circular icon of a character with binoculars is centered on the page. At the bottom, a message states "No images generated." To the right of the main content area is a sidebar titled "birdAI" with navigation links: Dashboard, Conversation, Image Generation (selected), Video Generation, Music Generation, Code Generation, and Settings. The main content area has a header "Image Generation" with the sub-instruction "Turn your prompt into an image". Below this is a text input field containing the prompt "A picture of an amphibious yordle holding a trident", followed by a "Generate" button. A small, semi-transparent circular icon of a character with binoculars is centered on the page. At the bottom, a message states "No images generated." To the right of the main content area is a sidebar titled "birdAI" with navigation links: Dashboard, Conversation, Image Generation (selected), Video Generation, Music Generation, Code Generation, and Settings.

birdAI

localhost:3000/image

Image Generation

Turn your prompt into an image

A picture of an amphibious yordle holding a trident

1 Photo 512x512 Generate

No images generated.

birdAI

Dashboard

Conversation

Image Generation

Video Generation

Music Generation

Code Generation

Settings

Image Generation

Turn your prompt into an image

A picture of an amphibious yordle holding a trident

1 Photo 512x512 Generate

Download Download Download

Download Download

Download

OUTCOME

The image shows three views of the birdAI web application interface, specifically the Video Generation section.

Left View: A screenshot of a browser window titled "birdAI" at "localhost:3000/video". The sidebar on the left has icons for Dashboard, Conversation, Image Generation, Video Generation (which is selected and highlighted in orange), Music Generation, Code Generation, and Settings. The main area is titled "Video Generation" with the sub-instruction "Turn your prompt into video". A text input field contains the prompt "A boy shouting!". Below it is a "Generate" button. A small illustration of a person looking through binoculars is centered. At the bottom, the message "No video generated." is displayed.

Middle View: A screenshot of the birdAI mobile application interface. The sidebar on the left lists "Dashboard", "Conversation", "Image Generation", "Video Generation" (selected and highlighted in orange), "Music Generation", "Code Generation", and "Settings". The main area is titled "Video Generation" with the sub-instruction "Turn your prompt into video". A text input field contains the prompt "A boy shouting!". Below it is a "Generate" button. A small illustration of a person looking through binoculars is centered.

Right View: A screenshot of the generated video frame. The title "Video Generation" is at the top, followed by the sub-instruction "Turn your prompt into video". The prompt "A boy shouting!" is shown in a text input field. Below is a large thumbnail image of a cartoon boy shouting, surrounded by colorful coral reef structures. A progress bar at the bottom indicates "0:00 / 0:03".

CONCLUSION

This Project is still in its early stages. The bird-AI project will exhibit the fundamental elements of a promising Gen AI endeavor.

It will demonstrate the potential for comprehensive, multimodal content generation by including varied datasets, API interaction, and a Transformer-based architecture. The bird-AI will aspire to soar even higher in the realm of AI innovation as it develops.

REFERENCES

- [1] <https://www.techtarget.com/searchenterpriseai/definition/generative-AI>
- [2] <https://www.nvidia.com/en-us/glossary/data-science/large-language-models/>
- [3] <https://en.wikipedia.org/wiki/GPT-1>
- [4] <https://openai.com/research/gpt-2-1-5b-release>
- [5] <https://openai.com/blog/gpt-3-apps>
- [6] <https://www.techtarget.com/searchenterpriseai/definition/generative-adversarial-network-GAN>
- [7] <https://openai.com/dall-e-2>