A Project Documentation on

AI-Powered Content Generation and Multimedia Utility Website

Submitted in fulfillment of the requirements of

the Semester VIII Subject of

FYP

in

Software Engineering

by

Omais Rana	**BSSE51F20S010**
Dawood Ilyas	**BSSE51F20S012**
Muhammad Farhan	**BSSE51F20S032**

Subject In-charge

Dr. Humaira Ijaz

Subject Supervisor

Dr. Humaira Ijaz

Department of SE

University of Sargodha

Table of Contents

Chapter One: Synopsis

1.1 Introduction

This project proposal highlights the development of a website that makes use of AI technology to generate high-quality content and offers various multimedia utilities which otherwise the person would have to search for on individual sites.

1.1.1 Background

In today's digital age, there is a high need to automate tedious tasks that take up much time and effort. Businesspersons, content creators, or even students actively seek out an efficient platform where they can get these small jobs done in a single place. People are continuously looking for ways to streamline their jobs and increase productivity in today's fast-paced environment where every second counts. People are more likely to choose platforms that not only save them valuable time but also excel in efficiency as the globe changes quickly. Time is money, and this adage is more accurate than ever.

1.1.2 Justification

Our website provides a one-stop shop for a variety of simple chores, such as image conversions or getting a written assignment done, rather than browsing through loads of websites and wasting valuable time. Users may easily find all these tools and information in our user-friendly library, which has been neatly organized. By doing this, we enable users to take back their precious time while also ensuring that the results produced are defined by great efficiency. Our technology is intended to be the ultimate time-saver, enabling users to focus on what matters most while maximizing their efforts.

1.2 Objectives

The primary goals of the "ContentCraft" project are as follows:

1.2.1 AI Content Creation

Create an AI-powered content generation tool that can generate text on various topics and analyze images i.e. marketing proposals, blog posts, describing pictures, and more.

1.2.2 Multimedia Utilities

Create a library of multimedia tools, such as file converters and code generators, to help users with their media needs.

1.2.3 User-Friendly Interface

Design a simple and user-friendly web interface using Bootstrap with Laravel as the main framework that allows users to access and use the provided tools efficiently.

1.2.4 Data Security

Ensure data privacy and security by implementing robust security measures to protect user-generated content and uploaded files.

1.3 Scope of Project

Our undertaking encompasses numerous key components, starting with AI Integration. This includes the incorporation of Generative Models and APIs to generate amazing textual content that intently emulates human language, improving the general person's experience. Additionally, we specialize in Multimedia Tools, wherein we are constructing a flexible library of multimedia utilities, which include but not limited to code creators and document converters. Simplifying the procedure of importing and downloading multimedia data while making sure of compatibility with an extensive variety of media formats. In the area of Web Development, we're making use of Bootstrap and vanilla CSS for the front-end, while leveraging PHP, Python, Node.JS and Express.JS for back-end implementation and seamless API integration. Lastly, our dedication to Data Security stays paramount, with sturdy measures in the area to defend users' records and documents, ensuring a stable and straightforward surrounding.
The following is a summary of the above-mentioned things.

AI Integration:

- Incorporation of Generative Models and APIs
- Generating text with precision

Multimedia Tools:

- Building a flexible library of multimedia utilities
- Simplifying multimedia data handling
- Ensuring compatibility with various media formats

Web Development:

- **Front-end technologies:** Vanilla CSS and Bootstrap
- **Back-end development:** PHP, Laravel, Python, Node.JS and Express.JS
- **Database:** MySQL

Data Security:

- Strong measures in place to protect user data and files
- Ensuring a secure and user-friendly environment

1.4 Methodologies Used (Techniques and Tools)

The development of "ContentCraft" will involve the following methodologies, techniques, and tools:

1.4.1 Machine Learning

Making use of machine learning algorithms for AI content generation, including pre-trained language models.

1.4.2 Front-End Technologies

Using Bootstrap for responsive and visually appealing front-end design.

1.4.3 Back-End Development

Implementing the back-end using Laravel framework. Node.JS and Express.JS used to handle API integration along with PHP libraries for tools integration while MySQL serves as database.

1.4.4 API Integration

Integrate third-party APIs where necessary to enhance the platform's functionality.

1.4.5 Incremental Approach

Enabling ourselves to deliver small portions of the product and complete our work in phases. This will reduce risk and provide us with better resource management.

1.4.6 Data Security

Implementing security measures to protect user data.

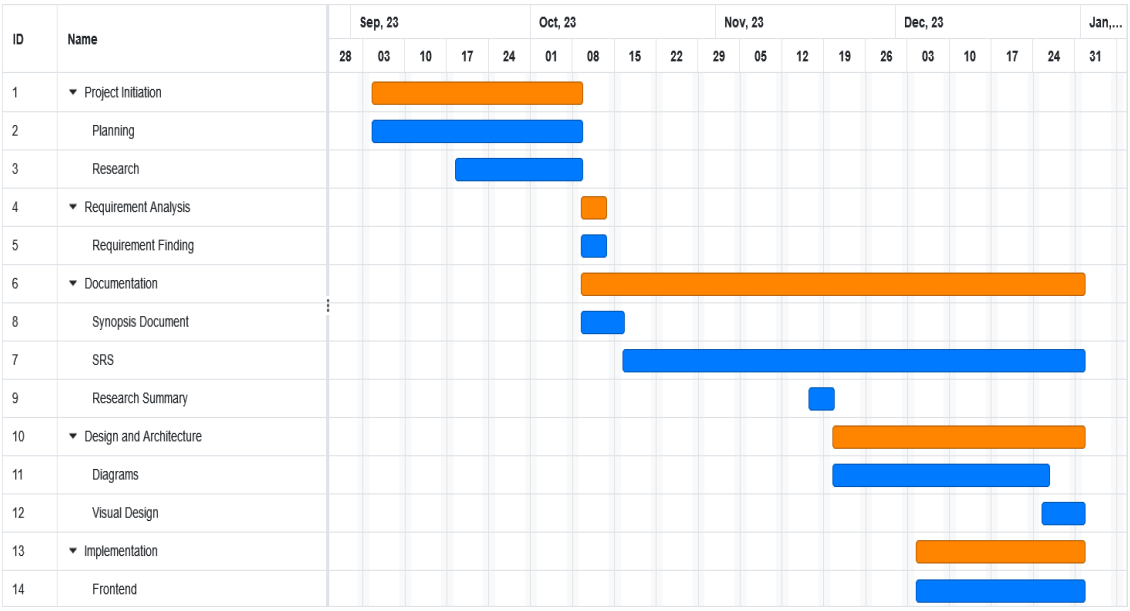
1.5 Expected Output

This project aims to give a user-friendly and efficient website that users will be able to access for their multimedia and content creation needs. It will give high-quality results while ensuring the users' data privacy. The project's success will depend on rigorous development, testing, and research.

1.6 Our Learning

Node.JS and Express.JS courses - Youtube
MERN Stack course - Coursera
API integration - Youtube
API understanding - Google Documentation
Multimedia understanding - Youtube
Generative AI course - Open Weaver
Laravel tutorials - Youtube
Python understanding - Youtube
PHP library integration - Youtube

1.7 Gantt Chart



1.8 Grammarly Report

Performance

Text score: 83 out of 100. This score represents the quality of writing in this document. You can increase it by addressing Grammarly's suggestions.

83

Word count

Characters	5,162	Reading time	3 min 2 sec
Words	760	Speaking time	5 min 50 sec
Sentences	68		

Readability

Metrics compared to other Grammarly users

Word length	5.6	Above average
Sentence length	11.2	Above average
Readability score	34	

Your text is likely to be understood by a reader who has at least some college education, but it may not be easy to read.

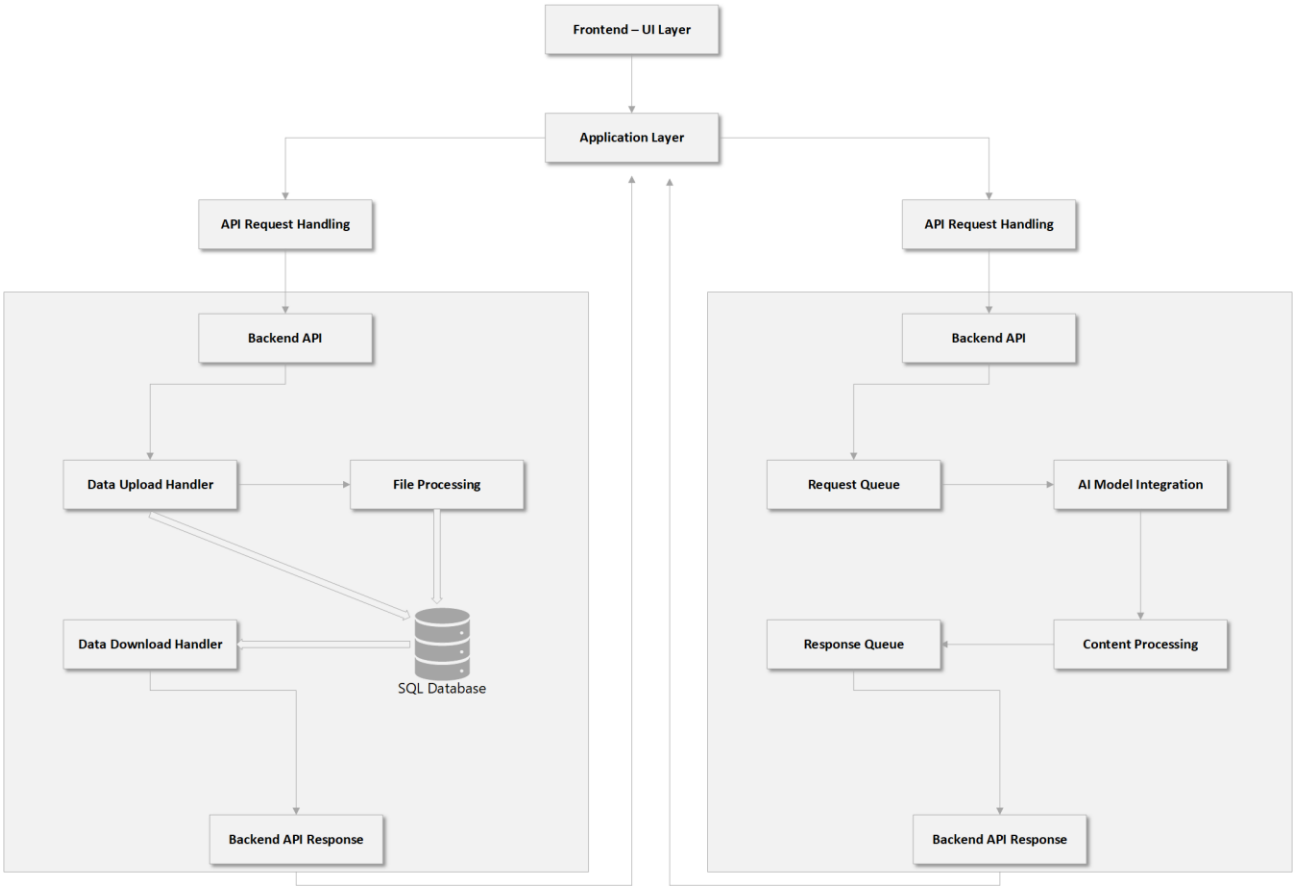
Chapter Two: Software Requirement Specification

2.1 Description

This document outlines the comprehensive blueprint for our innovative text-based content creation and multimedia featured website. Developed using Laravel at core and integrated with Google's API, the platform seamlessly combines advanced AI content generation with a user-friendly multimedia service. The SRS document delineates the system's functional and non-functional requirements, ensuring performance, reliability, security, and user-friendliness. It serves as a guiding document for development, encompassing both AI and multimedia modules.

2.2 Product Perspective

2.2.1 Block Diagram



2.2.2 Architecture

Detailed Architecture for AI Content Generation:

User Interface Layer:

- **Input** ← User gives text or image prompt.
- **Output** → Sends API requests for content generation.

Application Layer:

- **Input** ← Receives user prompts.
- **Output** → Sends API requests to the backend.

API Request Handling:

- **Input** ← Receives API requests.
- **Output** → Forwards requests to the backend API.

Backend API:

- **Input** ← Receives validated API requests.
- **Output** → Sends requests to the AI Model Integration.

Request Queue:

- **Input** ← Receives requests from the Backend API.
- **Output** → Manages a queue of requests for AI processing.

AI Model Integration:

- **Input** ← Receives requests from the Request Queue.
- **Output** → Connects to Google's API and forwards the request.

Content Processing:

- **Input** ← Request is processed, and content is generated.
- **Output** → Sends processed content for display.

Response Queue:

- **Input** ← Receives processed content from Content Processing.
- **Output** → Manages a queue of responses before delivering to the Backend API Response.

Backend API Response:

- **Input** ← Receives responses from the Response Queue
- **Output** → Sends responses back to the frontend for user display

Detailed Architecture for Multimedia Service:

User Interface Layer:

- **Input** ← User uploads data, such as files.
- **Output** → Sends API requests for operation.

Application Layer:

- **Input** ← Receives uploaded files.
- **Output** → Sends API requests to the backend.

API Request Handling:

- **Input** ← Receives API requests.
- **Output** → Forwards requests to the backend API.

Backend API:

- **Input** ← Receives API requests.
- **Output** → Sends requests to the File Upload Handler.

File Upload Handler:

- **Input** ← Receives requests and files from the Backend API.
- **Output** → Manages the upload process, validates, and stores files temporarily.

File Processing:

- **Input** ← Receives validated files.
- **Output** → Desired operation is performed.

File Storage:

- **Input** ← Receives files.
- **Output** → Stores the files securely in a designated location.

File Download Handler:

- **Input** → User requests for downloading processed files.
- **Output** ← Retrieves and delivers the files for user download.

Backend API Response:

- **Input** ← Receives responses from the File Download Handler.
- **Output** → Sends responses back to the frontend for user confirmation.

2.3 Specific Requirements

2.3.1 UI Requirements

- User friendly
- Easy prompt input
- Guideline box
- Easy file upload/download
- Clear response presentation
- Robust error handling

2.3.2 Hardware Reqs.

- Any device with web browser
- Processor (Dual-core or higher)
- RAM (Minimum 2GB)
- Internet connection

2.3.3 Software Reqs.

- Latest web-browsers
- JavaScript enabled

2.4 Functional Requirements

2.4.1 For AI module

- Prompts in varying length and formats
- Response coherent to inputs
- Response display (easy to read) on user device
- User can copy generated response to clipboard
- System will handle errors or limitations related to AI Model

2.4.2 For Multimedia-

- User can upload files and input data through user interface
- System will validate data format
- System will smoothly perform operation on given data
- User can download the files on local machine and use the data

2.5 Non-Functional Requirements

- Fast response time
- Web responsiveness
- Available 24/7
- Robustness
- Data privacy

2.6 Prioritization

Requirement prioritization has been based on a ranking method, where 1 indicates "not important" and 10 indicates "critical."

Performance: 10

- Performance optimization is vital for providing a smooth user experience.

Reliability: 9

- High reliability minimizes downtime and ensures a consistent user experience.

AI Content Generation: 9

- Optimizing AI models is essential for the core functionality of the website.

API and Integration: 8

- API and integration development is essential but may not be as critical as core features.

Data Management: 7

- Efficient data storage and retrieval are important for overall system performance.

User Interaction: 6

- User interaction features contribute to user engagement but may not be critical.

Security: 5

- Ensuring the security of user data is critical for user trust and compliance.

Moderation: 4

- Moderation is important for managing the platform.

Legal and Compliance: 3

- Compliance is important but may not require immediate focus unless there are specific regulations

2.7 Research Summary

PaLM API By Google

The PaLM API is a developer offering from Google that provides access to Google's large language models. It is a simple entry point and using it you can access the capabilities of Google's Generative AI models and develop AI-powered applications. It provides a non-complex and approachable way to start prototyping and experimenting with generative AI.

2.7.1 Text and chat services

These are two different ways to generate text using the PaLM API. The text service is for single-turn interactions, such as summaries or translations. The chat service is for multi-turn conversations, such as chatbots or tutors.

The high level client library for generative AI provides a set of functions that allow developers to use large language models for various natural language processing tasks. The **palm.generate_text** function allows the user to provide some initial text and have the model complete it with more

text. The **palm.chat** function allows the user to have a conversation with a model. The **palm.get_model** function allows the user to get information about a model by its name.

2.7.2 Embeddings service

This is a service that converts text into numerical vectors that capture semantic meaning and context. The embeddings can be used for tasks such as information retrieval, classification, or clustering.

2.7.3 Tuning service

This is a service that allows users to fine-tune the model behind the text service for a specific task. The tuning service requires a training dataset with examples of the desired task.

2.7.4 Model information

The PaLM API offers three different models optimized for specific use cases i.e. text-bison-001 model, chat-bison-001 model, embedding-gecko-001 model

1. **text-bison-001:** This is model is suitable for a variety of language tasks, such as extractive question answering, summarization, re-writing text in a different style. The maximum input tokens it can handle is 8192 and the maximum output tokens are 10242.
2. **chat-bison-001:** This model is fine-tuned for multi-turn conversation use cases like building a chatbot. It's designed to handle a maximum of 8192 input tokens and can generate a maximum of 1024 output tokens.
3. **embedding-gecko-001:** This model is used to generate embeddings, which are numerical representations of text that capture semantic meaning and context. The embedding-gecko-001 model takes a maximum of 3,072 input tokens and outputs 768-dimensional vector embeddings.

These models have attributes which are configurable parameters that can be set according to the use case.

Gemini API

As of December 2023, Google has replaced PaLM by an even powerful generative model called Gemini.

Here are some important details about the Gemini API:

Gemini API is a service that gives you access to the latest generative models from Google, such as Gemini Pro and Gemini Pro Vision. It has all the capabilities of PaLM and with added features.

Gemini models can accept text and image in prompts and output text responses. Some models, such as **Gemini Pro Vision**, can perform vision-related tasks, such as captioning or identifying images.

Gemini's forte in multi-modal tasks makes it a versatile tool for a variety of applications, especially those involving different data types.

Chapter Three: Use Cases

3.1 Prerequisite

Internet Connection:

- A stable internet connection is necessary to access the website.

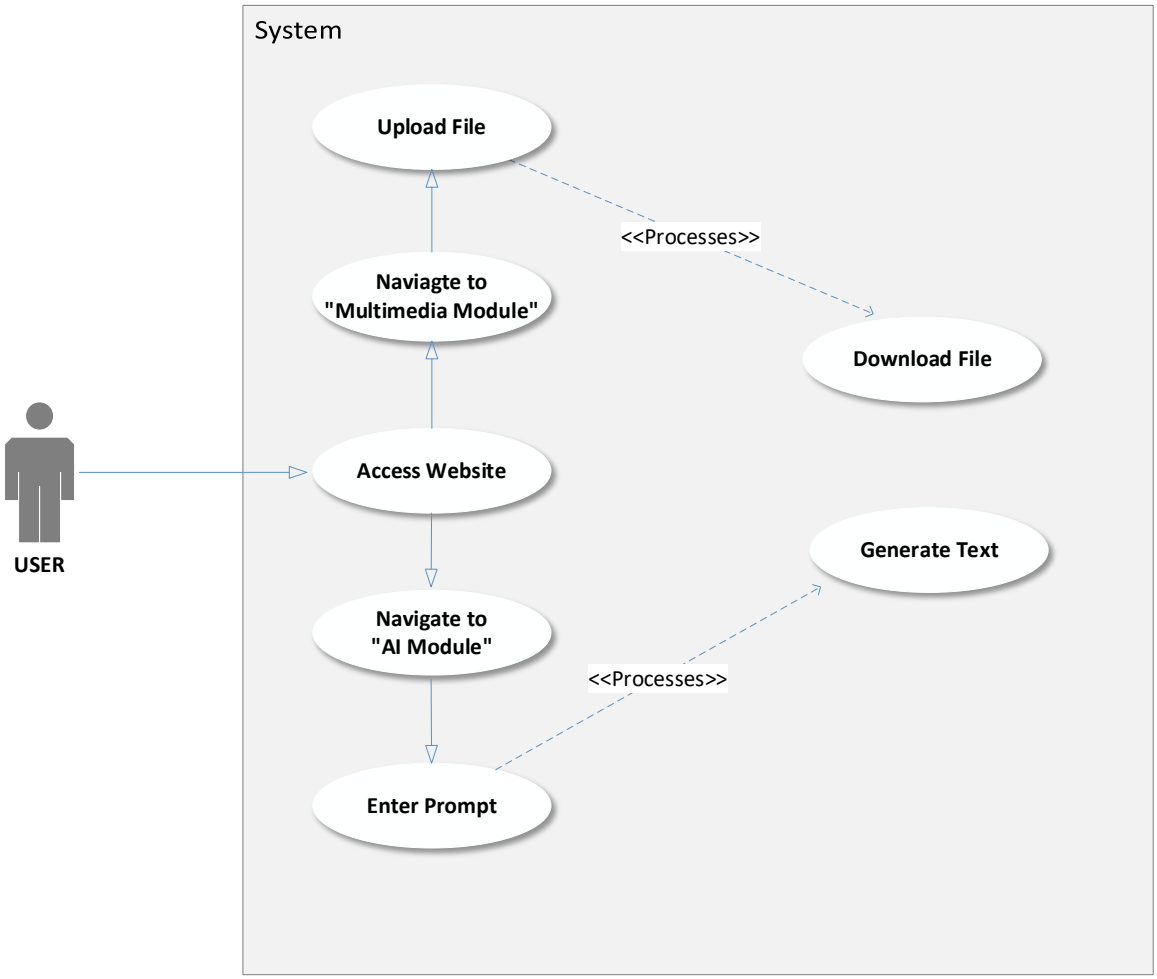
File Format Compatibility:

- Users should ensure that the files they upload are in supported formats.

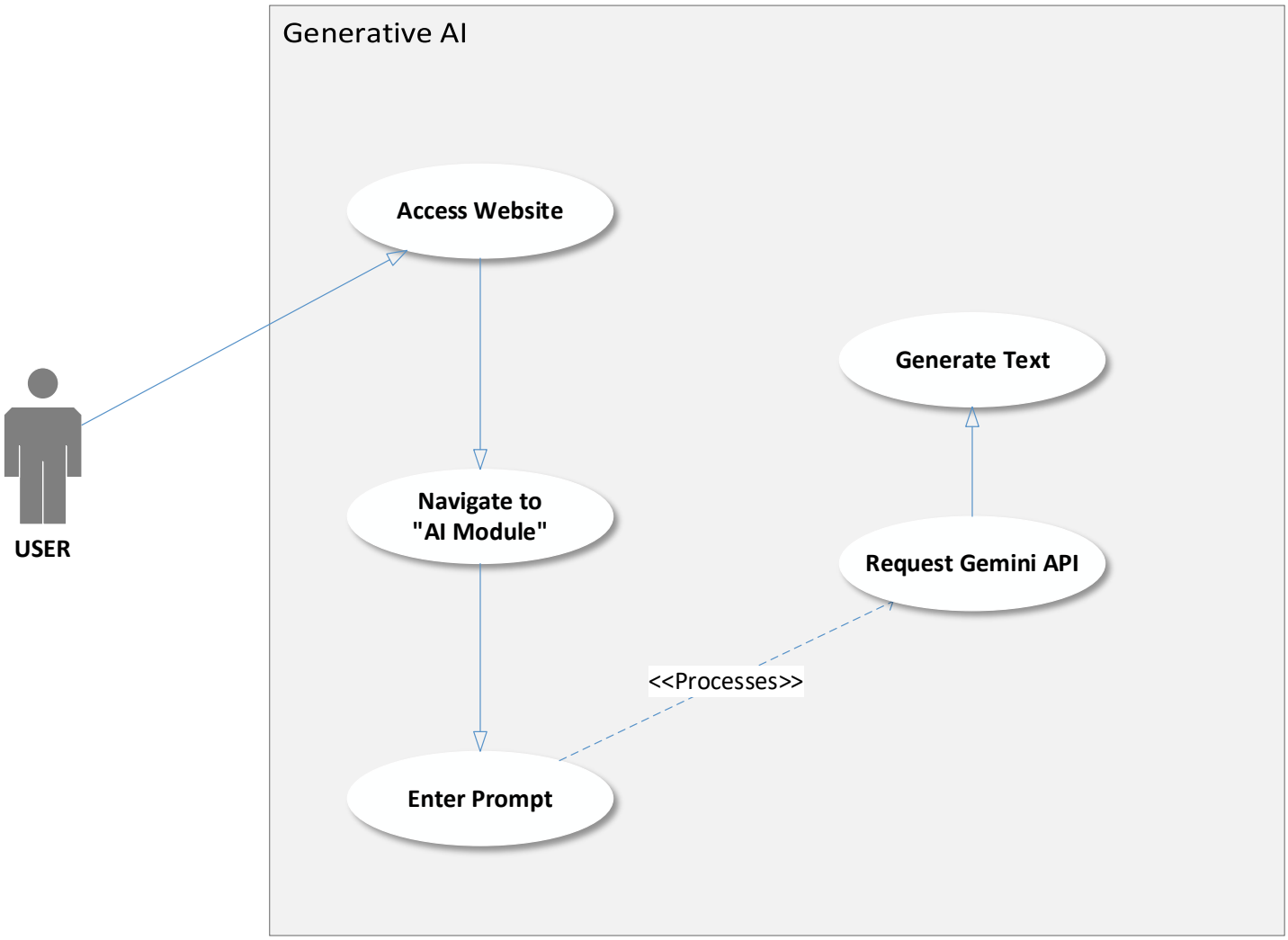
Understanding of Prompts

- Users should have a clear understanding of how to formulate effective prompts when using the Generative AI module to achieve desired and relevant content outcomes.

3.2 General Use Case Diagram



3.3 AI Module



3.3.1 Use Case 1: Generating Product Descriptions

User Action:

1.User accesses the website and navigates to the "Generative AI" section.

2.Enters a prompt like "Create a compelling product description for a new tech gadget."

System Response:

1.Generative AI processes the prompt using Google API.

2.Generates a detailed and engaging product description based on the user's input.

3.3.2 Use Case 2: Creating Taglines for Marketing Campaigns

User Action:

1.User navigates to the "Generative AI" section.

2.Provides a prompt like "Generate catchy taglines for a fitness brand."

System Response:

1.Generative AI, powered by Google API, processes the prompt.

2.Delivers a list of creative and impactful taglines tailored to the fitness brand.

3.3.3 Use Case 3: Describing an Image

User Action:

1.User accesses the "AI Image Analysis" section.

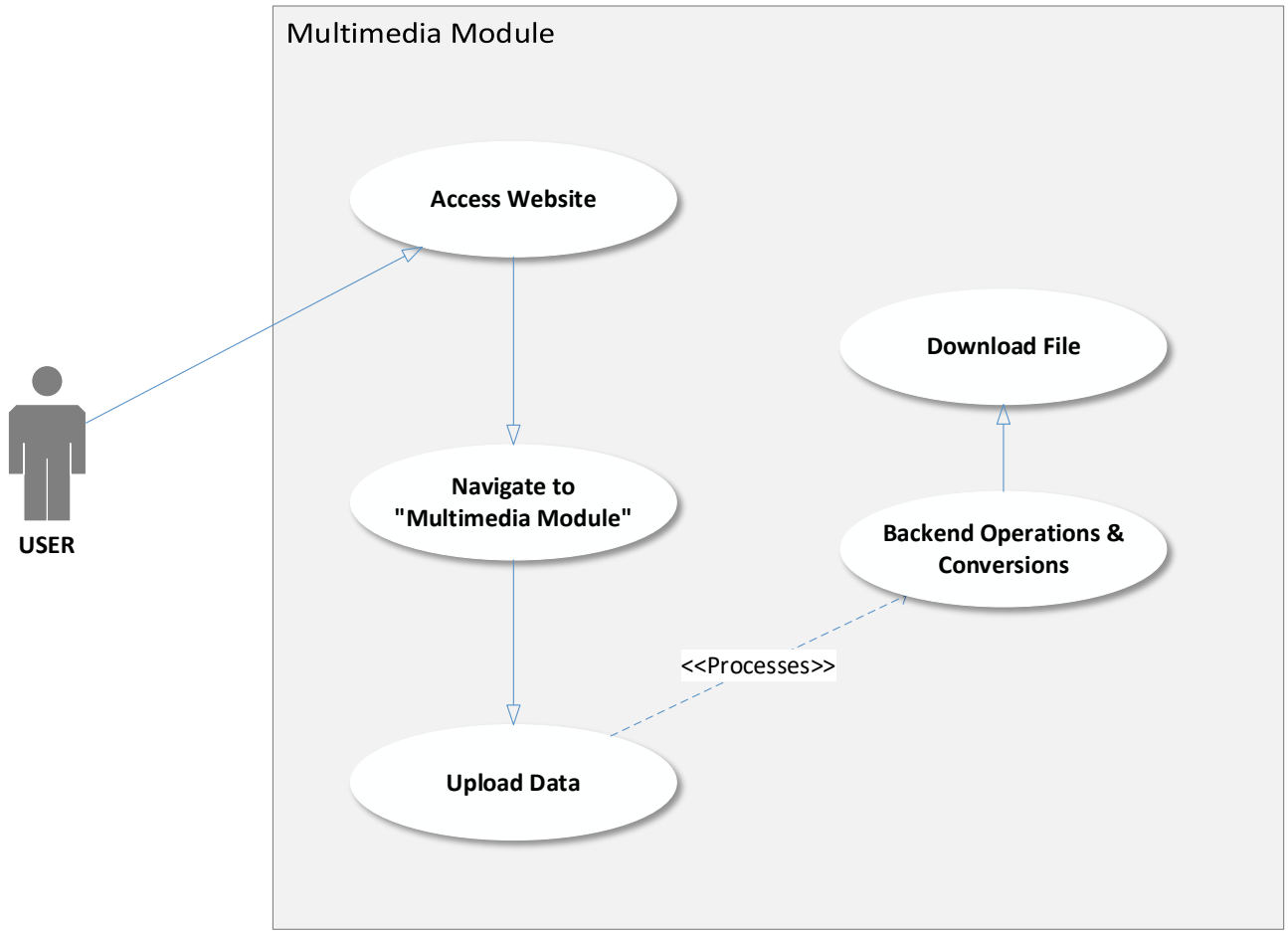
2.Uploads an image as prompt.

System Response:

1.Google API sends the image for processing to understand it.

2.Gives a text describing the context of the image.

3.4 Multimedia System



3.4.1 Use Case 1: File Conversion

User Action:

1.User switches to the "Multimedia" section and selects “File Conversion”.

2.Uploads a Word document with the product details.

System Response:

1.File Conversion module processes the Word document.

2.Converts the document to various formats like PDF, and TXT.

3.4.2 Use Case 2: QR Code Generation

User Action:

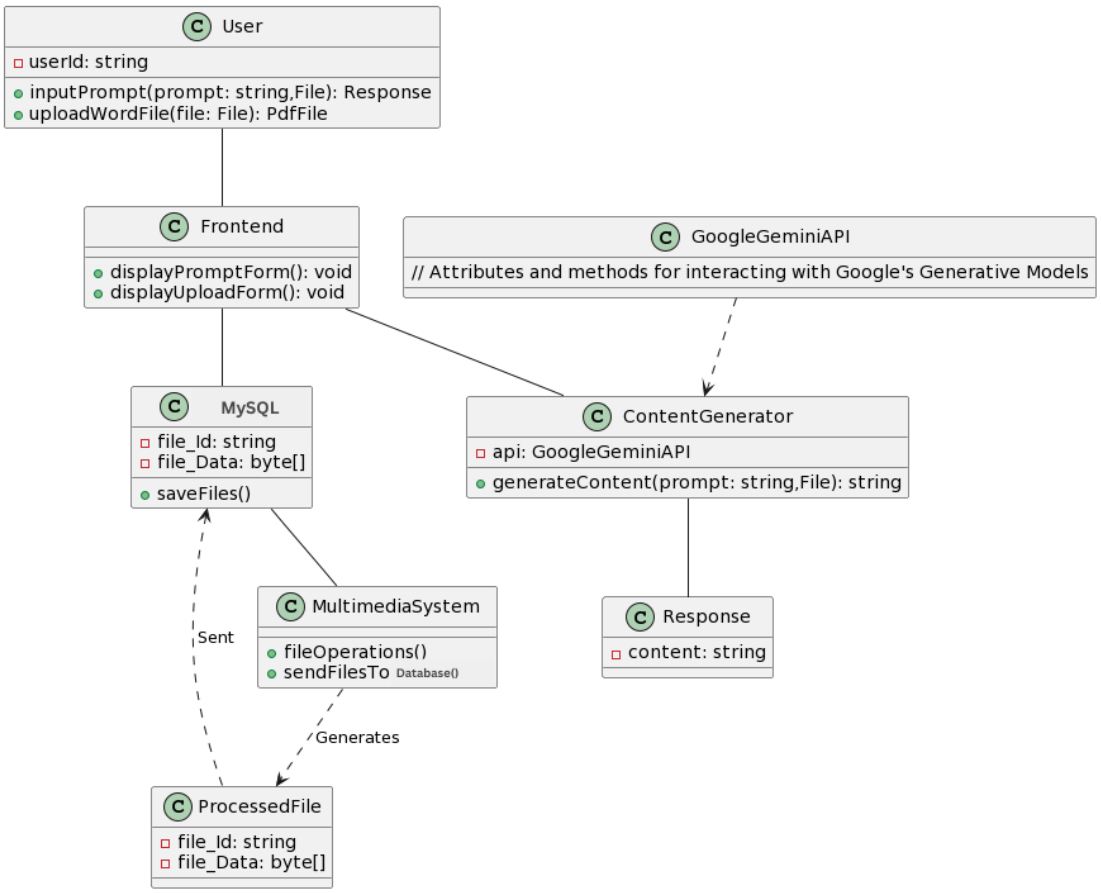
1.User switches to the "Multimedia" section and selects “QR Code Generator”.

2.User enters the desired destination of the QR Code.

System Response:

1.System will generate a QR Code and user can download it in SVG format.

3.5 Class Diagram

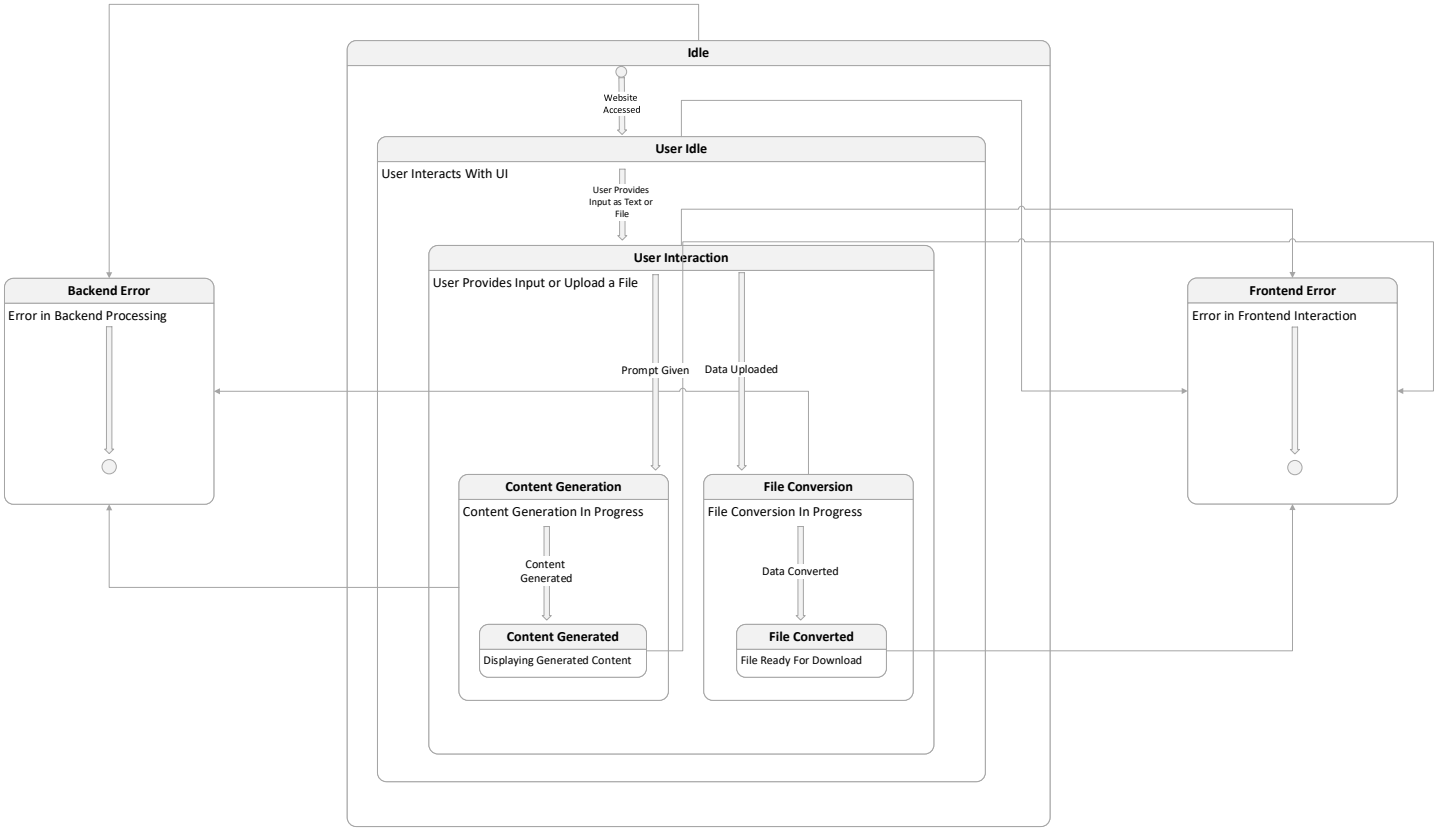


3.6 State Transition Table

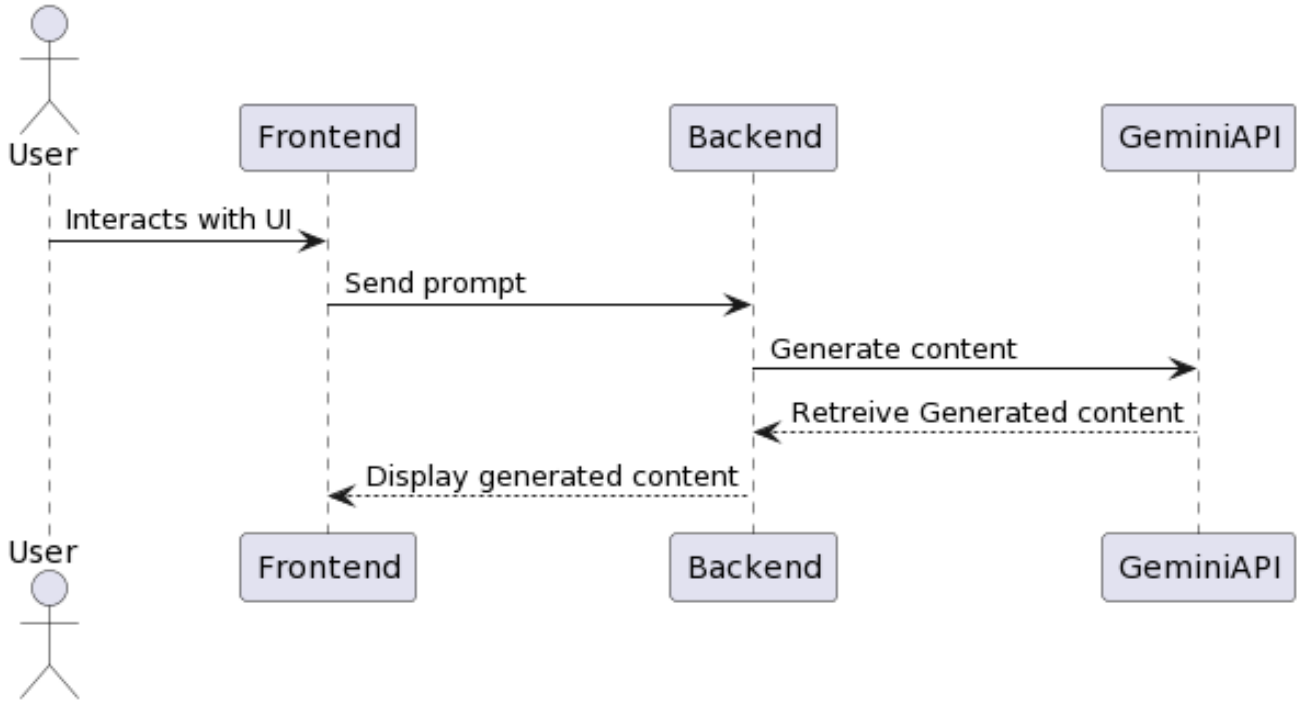
StateTransition

Current State	Event	Next State	Action
——--	–	—	—
Idle	Website Accessed	User_Idle	-
User_Idle	User interacts	User_Interacting	-
User_Interacting	User provides input	ContentGeneration	-
User_Interacting	User uploads file	FileConversion	-
ContentGeneration	Content generated	ContentGenerated	Display generated content
FileConversion	File converted	FileConverted	Display converted file
ContentGenerated	User continues	User_Idle	-
FileConverted	User continues	User_Idle	-
FrontendError	Error in interaction	-	Display error message
BackendError	Backend error	-	Display error message

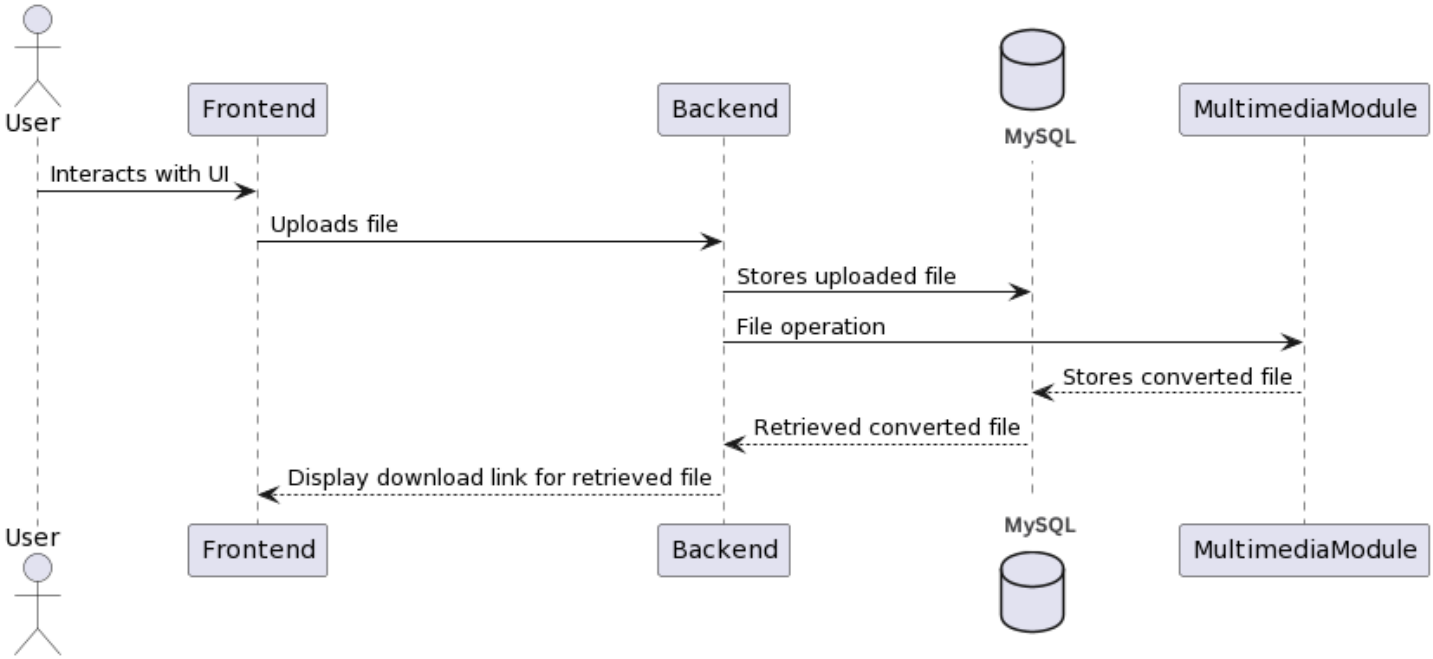
3.7 State Transition Diagram



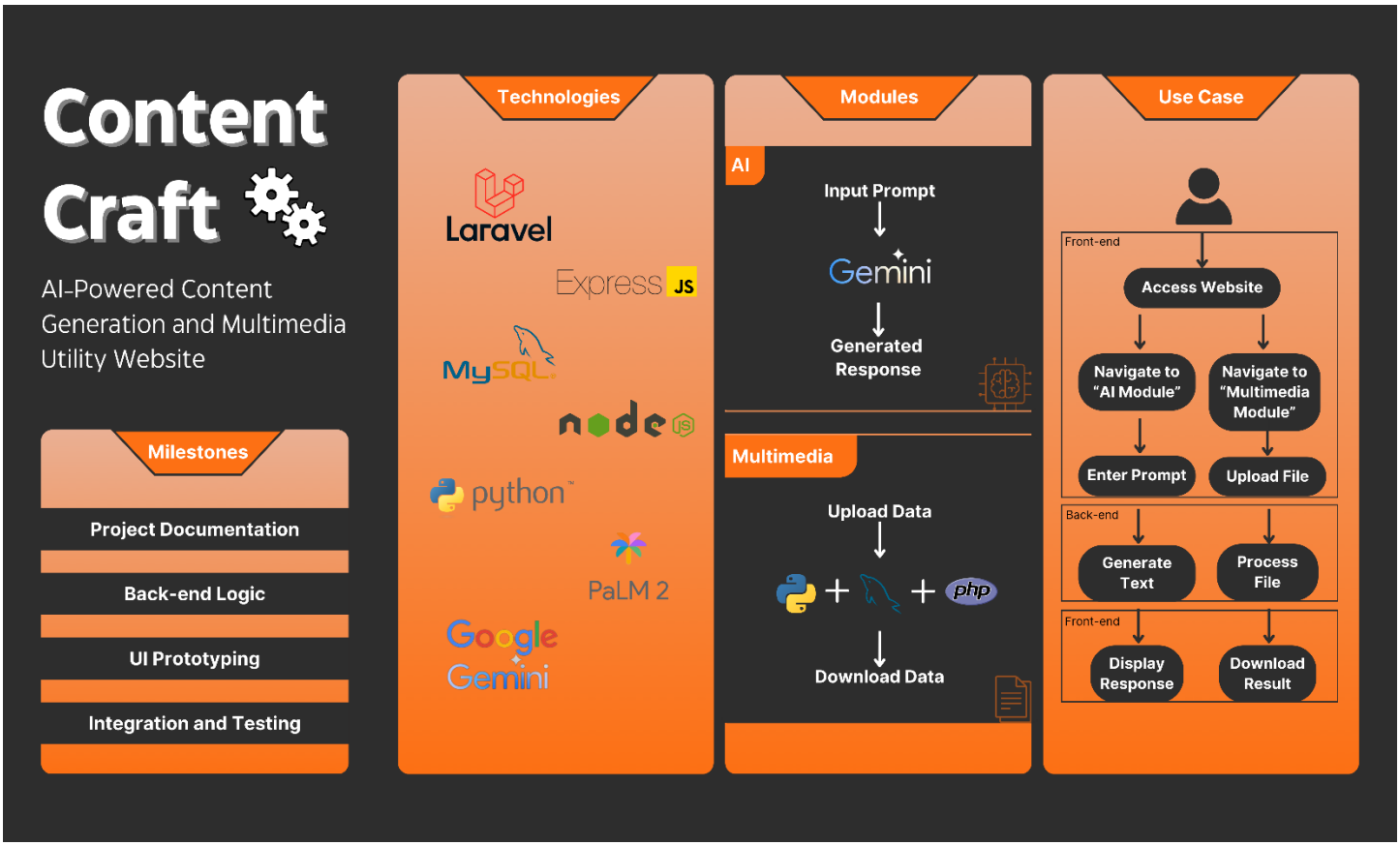
3.8 Sequence Diagram for AI Module



3.9 Sequence Diagram for Multimedia Module



3.10 Graphical Abstract



Chapter Four: Test Plan

4.1 Introduction

This test plan outlines the strategy for testing the ContentCraft web platform. The objective is to ensure the platform functions as intended, delivers high-quality results, and provides a user-friendly experience.

4.2 Scope of Testing

The testing scope will encompass the following key areas:

- **Functionality Testing:** Verify that all platform features, including AI content generation, multimedia utilities and user interface elements work correctly.
- **Usability Testing:** Evaluate user experience with the platform's interface, navigation, and overall ease of use.
- **Performance Testing:** Assess the platform's loading times and responsiveness across different devices.

4.3 Testing Strategy

4.3.1 Unit Testing

Focus: Isolating and testing individual software components (e.g., AI content generation modules, file conversion utilities).
Participants: Testing Team (Omais Rana, Dawood Ilyas)
Techniques:

- White-box testing: Examining internal code structure to ensure proper functionality.

4.3.2 Integration testing

- **Focus:** Verifying seamless interaction between different platform components.

Participants: Testing Team (Omais Rana, Dawood Ilyas)

Techniques:

- Bottom-up approach: Testing low-level modules individually before integrating them into larger systems.
- API testing tools: Verifying communication between platform components through APIs.

4.3.3 System Testing

Focus: Evaluating the entire platform to ensure it meets user requirements and functions as a cohesive system.

Participants: Testing Team (Omais Rana, Dawood Ilyas)

Techniques:

- Black-box testing: Testing platform functionalities from a user's perspective without examining internal code.
- Exploratory testing: Unstructured testing to identify unexpected issues or edge cases.
- Regression testing: Re-running previously successful tests after code modifications to ensure no regressions occurred.

4.3.4 Acceptance Testing

Focus: Involving real users to evaluate the platform's functionality, usability, and overall user experience.

Participants: Peer (Mujtaba Jawadi)

Techniques:

- User scenario testing: Users walk through typical tasks they would perform to assess workflow efficiency.
- Usability testing: Observing user interaction with the platform to identify areas for improvement.
- A/B testing: Presenting different versions of UI elements to users to determine which performs better.

4.4 Test Environment

A dedicated testing environment will be set up to mirror the production environment as closely as possible.

4.5 Test Case Design and Description

Test Case ID:	TC-AI-001	**Test Case Name**	AI Text Generation Test
Test Date:	25-5-2024	**Tested By**	Omais Rana
Objective	Verify the AI content generation module produces grammatically correct and relevant text content.		
Pre-Requisites	• AI content generation module is functional. • Sample user input data is prepared (e.g., keywords, desired content length).		
Assumptions:	The module will generate accurate results and high quality content.		
Steps	• Provide user input data (prompt). • Initiate the AI content generation process. • Review the generated content.		
Expected Results	• The generated content is free of grammatical errors and adheres to proper sentence structure, as verified by a grammar checker or through manual review. • The content aligns with the chosen topic and demonstrates a relevant understanding of the subject matter.		
Pass/Fail Criteria	• **Pass**: The generated content meets all expected results. • **Fail**: The generated content contains grammatical errors, is irrelevant to the topic, or demonstrates a lack of understanding of the subject matter.		
Comments This test case verifies the correct functioning of the AI content generation.			

Test Case ID:	TC-AI-002	**Test Case Name**	AI Image Analysis Test
Test Date:	25-5-2024	**Tested By**	Dawood Ilyas
Objective	Verify the AI Image Analysis module produces relevant text according to the given image and prompt.		
Pre-Requisites	• AI image analysis module is functional. • Sample user input data is prepared (e.g. image, keywords, desired content length).		
Assumptions:	With a clear prompt and image data, the system will generate accurate results.		
Steps	• Provide user input data (image and prompt). • Initiate the AI image analysis process. • Review the generated text.		
Expected Results	• The generated text aligns with the given prompt and chosen image, showing understanding of the data. • The generated content is free of grammatical errors.		
Pass/Fail Criteria	• **Pass**: The generated content meets all expected results. • **Fail**: The generated content contains grammatical errors, is irrelevant to the topic, or demonstrates a lack of understanding of the subject matter.		
Comments This test case verifies the correct functioning of the AI image analysis.			

Test Case ID:	TC-MM-003	**Test Case Name**	Database Test
Test Date:	25-5-2024	**Tested By**	Omais Rana
Objective	Test the interaction between the user interface and the database when generating results by making use of URL Shortener functionality.		
Pre-Requisites	• Database is operational. • Sample user input data is prepared (valid url).		
Assumptions:	The database will store a valid shortened url and display it on the user interface.		
Steps	• Provide a valid url. • Initiate the shortening process • Review the generated url.		
Expected Results	• The database generates a shortened url and displays it on the user interface. • The generated url is works.		
Pass/Fail Criteria	• **Pass**: The generated url is accurately showing and working. • **Fail**: The database fails to provide the url or the url is invalid.		
Comments This test case verifies the correct functioning of the database along with url shortener functionality.			

Test Case ID:	TC-MM-004	**Test Case Name**	File Conversion Test
Test Date:	25-5-2024	**Tested By**	Dawood Ilyas
Objective	Verify the PDF-to-Word feature of multimedia module works correctly.		
Pre-Requisites	• PDF-to-Word feature is functional. • Sample user input file is prepared.		
Assumptions:	The module will convert the file successfully.		
Steps	• Provide user input file. • Initiate the conversion process. • Review the converted word file.		
Expected Results	• The converted file is in Word format. • There is no problem downloading the converted file and reading it afterwards.		
Pass/Fail Criteria	• **Pass**: File conversion meets all expected results. • **Fail**: File conversion fails, gives error or the file is non-readable.		
Comments This test case verifies the correct functioning of the file conversion feature.			

Test Case ID:	TC-UT-005	**Test Case Name**	Responsiveness and Usability Test
Test Date:	26-5-2024	**Tested By**	Mujtaba Jawadi
Objective	Evaluate the usability and responsiveness of the ContentCraft platform across different devices.		
Pre-Requisites	• The platform is operational on different screens for testing.		
Assumptions:	User has a basic understanding of web browsing and online platforms.		
Steps	• Observe user completing the defined tasks on a desktop screen. • Monitor their interaction with the interface, noting any difficulties or areas of confusion. • Repeat the defined tasks on a mobile screen. Evaluate the usability and responsiveness of the platform in a portrait and landscape orientation.		
Expected Results	• User can successfully complete the defined tasks on all device screens with minimal difficulty. • The user interface is intuitive and adapts seamlessly across different screen sizes. • User provides positive feedback on the platform's usability and overall user experience.		
Pass/Fail Criteria	• **Pass**: User can complete at least 80% of the defined tasks on each device without major obstacles. They provide positive feedback on the usability and responsiveness of the platform. • **Fail**: User encounters significant difficulties completing tasks on any device. The interface is not intuitive or responsive across different screen sizes. User feedback highlights negative experiences.		
Comments With this test ContentCraft can ensure a user-friendly platform that caters to a wider audience and provides a seamless experience on any device.			

Sr. no.	Test Case ID	Test case name	Status	Date Completed
1	TC-AI-001	AI Content Generating Test	Completed	25-2-2024
2	TC-AI-002	AI Image Analysis Test	Completed	25-5-2024
3	TC-MM-003	Database Test	Completed	25-5-2024
4	TC-MM-004	File Conversion Test	Completed	25-5-2024
5	TC-UT-005	Usability and Responsiveness Test	Completed	26-5-2024

4.7 Functions To be Tested

4.6.1 Core Functionalities:

- **AI Content Generation:**
    - Text generation for various content types (e.g., proposals, blog posts, image descriptions).
    - Quality and relevance of generated content.
- **Multimedia Utilities:**
    - Functionality of each utility tool (e.g., file conversion, code generation).
    - Integration and compatibility with various file formats.
    - Output accuracy and quality of generated content from utilities.

4.6.2 User Interface & User Experience:

- **Navigation and ease of use:** Can users easily find and utilize platform functionalities?
- **Intuitive interface design:** Is the interface visually appealing and user-friendly?
- **Responsiveness across devices:** Does the platform function optimally on desktops, tablets, and mobile devices?

4.7 Risks

- **AI Model Performance:** The AI content generation model might not perform as expected, leading to inaccurate or irrelevant content.
- **Third-Party API Integration:** Integration with third-party APIs could encounter compatibility issues or unexpected behavior.
- **Database Problems:** Database might not work as expected or crash with user load.
- **Performance Issues:** Processing large or complex multimedia files could lead to performance problems, causing delays or crashes within the platform.
- **Usability Issues:** The user interface might not be intuitive, hindering user experience and adoption.

4.9 Success Criteria

The testing process will be considered successful when:

- All critical functionalities operate as designed.
- Usability testing reveals a user-friendly and intuitive platform.
- Performance testing demonstrates acceptable loading times.
- Security testing identifies and resolves any critical vulnerabilities.
- User acceptance testing confirms the platform meets user needs.