RankMorph – AI Based SEO Analyzer

BY

Md. Sakib Khandaker

ID: 191-16-397

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computing and Information System

Supervised By

Md. Mehedi Hassan

Lecturer (Senior Scale)

Department of CIS

Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY

12 JANUARY 2025

APPROVAL

This Project titled **"RankMorph – AI Based SEO Analyzer"**, submitted by **Md. Sakib Khandaker**, ID: 191-16-397 to the Department of CIS, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in CIS and approved as to its style and contents. The presentation has been held on 12-01 -2025.

BOARD OF EXAMINERS

12.01.25

Md Sarwar Hossain Mollah — **Chairman**
Associate Professor and Head
Department of Computing & Information Systems
Faculty of Science & Information Technology
Daffodil International University

12.01.25

Md. Nasimul Kader — **Internal Examiner**
Assistant Professor
Department of Computing & Information Systems
Faculty of Science & Information Technology
Daffodil International University

12.01.25

Md. Mehedi Hassan — **Internal Examiner**
Lecturer (Senior Scale)
Department of Computing & Information Systems
Faculty of Science & Information Technology
Daffodil International University

Anowar Hossain — **External Examiner**
Founder & CEO at DocTime

DECLARATION

We hereby declare that, this project has been done by us under the supervision of **Md. Mehedi Hassan, Lecturer (Senior Scale), Department of CIS,** Daffodil International University. I also declare that neither this project nor any part of this project has been submitted elsewhere for the award of any degree or diploma.

Supervised by: Md. Mehedi Hassan

12.07.25

Lecturer (Senior Scale)
Department of CIS
Daffodil International University

Submitted by: Md. Sakib Khandaker

ID: 191-16-397
Department of CIS
Daffodil International University

ACKNOWLEDGEMENT

First, I express my heartiest thanks and gratefulness to almighty Allah for His divine blessing making us possible to complete the final year project successfully.

I am grateful and wish my profound indebtedness to **Md. Mehedi Hassan, Lecturer (Senior Scale), Department of CIS Daffodil International University, Dhaka.** The Deep knowledge & keen interest of my supervisor in the field of "Web development" helped to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting them at all stages have made it possible to complete this project.

I would like to express my heartiest gratitude to **Md Sarwar Hossain Mollah**, **Head,** Department of CIS, for his kind help to finish my project and also to other faculty members and the staff of the CIS department of Daffodil International University.

I would like to thank my entire course mate in Daffodil International University, who took part in this discuss while completing the course work.

ABSTRACT

You may read the findings that go along with my concept, a "RankMorph – AI Based SEO Analyzer," in full. This article discusses in depth the techniques utilized to transform the idea into a working website. The client dashboard is one feature that is very noticeable to system users. I'm contacting you to provide a project proposal that fits your needs and entails producing SEO feedback using the website data you've given. We'll build a system that uses AI to generate SEO feedback reports. The following features will be supported by the project: Client-side: Customers can directly contribute text. I will transform HTML data to text if it is supplied. API Integration: I will extract and process data if an API is accessible. A chat GPT-based AI model will receive the client's provided text or HTML and use it to generate SEO feedback in response to preset system prompts. The system will only be accessible to authorized users who can log in. highlighting the team's specifics and the goal of the endeavor. provide a form or contact information for questions. Put in place a login procedure to guarantee that only people with permission may access the site. The client will either utilize the API for URL-based input or supply text and HTML files for conversion. If necessary, convert HTML input to clean text. For SEO comments, send the converted text to GPT. Display the AI-generated comments in a readable manner. Add polished static pages for Contact Us and About Us. Make sure the system is easy to use, with intuitive navigation and features. accepts text, HTML, and API data forms uses AI to automate the SEO feedback process. Limits access to just those who are permitted. Contains more pages to improve the user experience. Kindly inform me whether this proposal satisfies your requirements. I can't wait to work with you to make this idea a reality. If you would want to schedule a call to talk more, please let me know. title of this entire document. For the frontend backend, I have used html, css, python. To set up my system application, all you need is a desktop computer and internet access; costly software or computer components are not required. You may utilize my platform as a worldwide repository and as easily accessible apps from anywhere if you have the right login credentials. With a typical internet connection, practically any user may use my platform-neutral solution at any time and from any location. I could also modify my system to meet specific needs.

TABLE OF CONTENTS

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Introduction

1.1 Introduction

An application that shows the relationships and interactions between other applications is called a system. On computers, the "System" page contains connections to applications, programming, and tools for system management. Although the word "system" may signify different things according to the context, the idea is basically the same. A complete framework is created by the "RankMorph – AI Based SEO Analyzer " by the integration of several technologies. This framework's several modules implement the restrictions outlined for each system in question. There are several systems in each module. I'm reaching out to you via this web-based application to offer a project proposal that meets your requirements and involves generating SEO feedback according to the website data you've provided. We'll create a system that generates SEO feedback reports using artificial intelligence.

In the competitive digital world of today, improving online exposure and website performance need efficient search engine optimization, or SEO. In order to meet this demand, I suggest creating an AI-powered system that uses supplied website data to automatically generate SEO feedback. This cutting-edge technology will provide customers the option to contribute data via API integration, HTML file uploads, or manual entry. Superior GPT-based AI technology will be used by the system to evaluate the input and produce useful SEO insights. With its user-friendly design, encrypted registration process, and extra static pages for assistance and context, this system is designed to offer a professional, safe, and effective way to boost a website's SEO performance. As companies work to improve their online visibility, SEO, or search engine optimization, has emerged as a crucial component of success. However, it frequently takes a great deal of skill and physical labor to produce intelligent and useful SEO feedback. I suggest creating a sophisticated AI-powered SEO feedback system that makes use of GPT-based AI models in order to expedite this procedure. Clients will be able to provide data via this system in a variety of formats, such as HTML files, human text entry, or API-based URL extraction. After processing, the AI will examine the data and produce thorough SEO feedback reports based on preset prompts. Only those with special permission can access the system thanks to the platform's secure authentication process. Furthermore, adding expertly

created static pages—like an About Us and Contact Us page—will improve user experience and offer the required background information and assistance. For companies of all sizes, this system seeks to provide an inventive way to streamline and improve SEO procedures by fusing flexibility, efficiency, and security.

CHAPTER 2

Initial Study

2.1 Project Proposal

Objectives

The primary aim of the website is to provide the competitive digital world of today, improving online exposure and website performance need efficient search engine optimization, or SEO. In order to meet this demand, I suggest creating an AI-powered system that uses supplied website data to automatically generate SEO feedback. This cutting-edge technology will provide customers the option to contribute data via API integration, HTML file uploads, or manual entry. Superior GPT-based AI technology will be used by the system to evaluate the input and produce useful SEO insights. Beyond this, I may strive to achieve the following:

- Create a safe, AI-driven system that uses website data to produce SEO feedback reports.
- Offer several ways for users to submit data, such as API connectivity, HTML file uploads, and manual text entry.
- Analyze text using GPT-based AI models to produce useful SEO insights depending on preset prompts.
- By limiting access to the platform to authorized users only, an authentication system may provide safe access.
- Create processes that are easy to use and navigate for smooth operation.
- Transform HTML inputs into clear text so the AI model can process them efficiently.
- Provide SEO feedback produced by AI in a clear, polished manner.
- Add static pages, including a Contact Us page for questions and an About Us page outlining the goal of the initiative.

Benefits of the website:

- Develop a secure, AI-powered system to generate SEO feedback reports based on website data.

- Provide multiple data submission methods, including manual text entry, HTML file uploads, and API integration.
- Utilize GPT-based AI models to analyze text and generate actionable SEO insights tailored to predefined prompts.
- Ensure secure access through an authentication system that restricts the platform to authorized users only.

2.2 Background

The success of online enterprises is greatly influenced by Search Engine Optimization (SEO), which raises user engagement, increases website exposure, and draws in organic traffic. As the need for efficient SEO tactics increases, companies frequently look for professional advice to maximize their online content. It takes a lot of effort, knowledge, and resources to use traditional SEO analysis techniques. With the development of artificial intelligence (AI), in particular natural language processing (NLP) models such as GPT, this process may now be automated, making it more accessible and effective for businesses [1] [2].

By creating an AI-powered SEO feedback system that automates the study and reporting of website data, this initiative seeks to close the gap. The solution simplifies what was before a difficult and resource-intensive operation by using GPT-based AI technology to produce comprehensive, actionable feedback on SEO performance [3]. For companies looking to improve their digital footprint effectively, this solution promises to be revolutionary since it integrates secure access, flexible input methods, and an intuitive design [2][3].

2.3 Problem Area

- Time-consuming SEO Analysis: Conventional techniques for producing SEO feedback need a lot of resources and professional assistance.
- Absence of Automation: For companies that manage several websites or sizable datasets, manual SEO analysis is inefficient and not scalable.
- Inconsistent Quality: Human analysis variability might result in feedback that is both inconsistent and lacking.

- Restricted Accessibility: Because of financial and resource limitations, small and medium-sized enterprises might not have easy access to expert SEO services.
- Problems with Data Formats: Websites frequently contain a variety of data formats (text, HTML, API), which makes processing and analysis difficult.
- Security Issues: There may be security hazards associated with unfettered access to private website data.
- User Experience Gaps: Current tools frequently fall short in terms of having an intuitive user interface or easy navigation.

2.4 Possible Solution

- AI-Powered Automation: Generate feedback and analyze SEO automatically with GPT-based AI.
- Multi-Format Data Support: Provide flexible input via text, HTML files, and API connectivity.
- Simplified Processing: Put strong algorithms into place to transform HTML into clear text so that it can be analyzed efficiently.
- Enable only authorized users to access the platform by integrating an authentication mechanism.
- Provide consistent, accurate, and actionable SEO analytics by using AI models and pre-established prompts.
- User-Friendly Interface: Create a platform with easy-to-use navigation and straightforward processes.
- Scalability: Create a system that can effectively manage big datasets and numerous users.
- For user interaction and support, including static support pages such as About Us and Contact Us.

CHAPTER 3

Literature Review

3.1 Discussion on problem domain based on published articles

The difficulties posed by conventional SEO procedures underscore the urgent need for creative fixes. Numerous studies have shown that SEO is still a crucial component of digital marketing as it has a direct impact on search engine rankings and website exposure. However, small companies find it challenging to successfully compete with manual techniques to SEO feedback, which are labor-intensive and frequently need a large investment of time and expertise [1] [2].

In order to overcome these obstacles, automation in SEO analysis has showed potential. After being trained on large datasets, AI models like GPT may produce informative feedback, which makes them useful tools for enhancing website performance. According to published research, AI-powered systems may reduce human error and standardize feedback, providing scalable and reliable solutions [3] [4]. Another major challenge is the variety of formats used to store website data, including text, HTML, and material available through APIs. In addition to saving time, automated systems that can process and analyze these formats provide accessibility for companies with a range of technological specialties [2] [5].

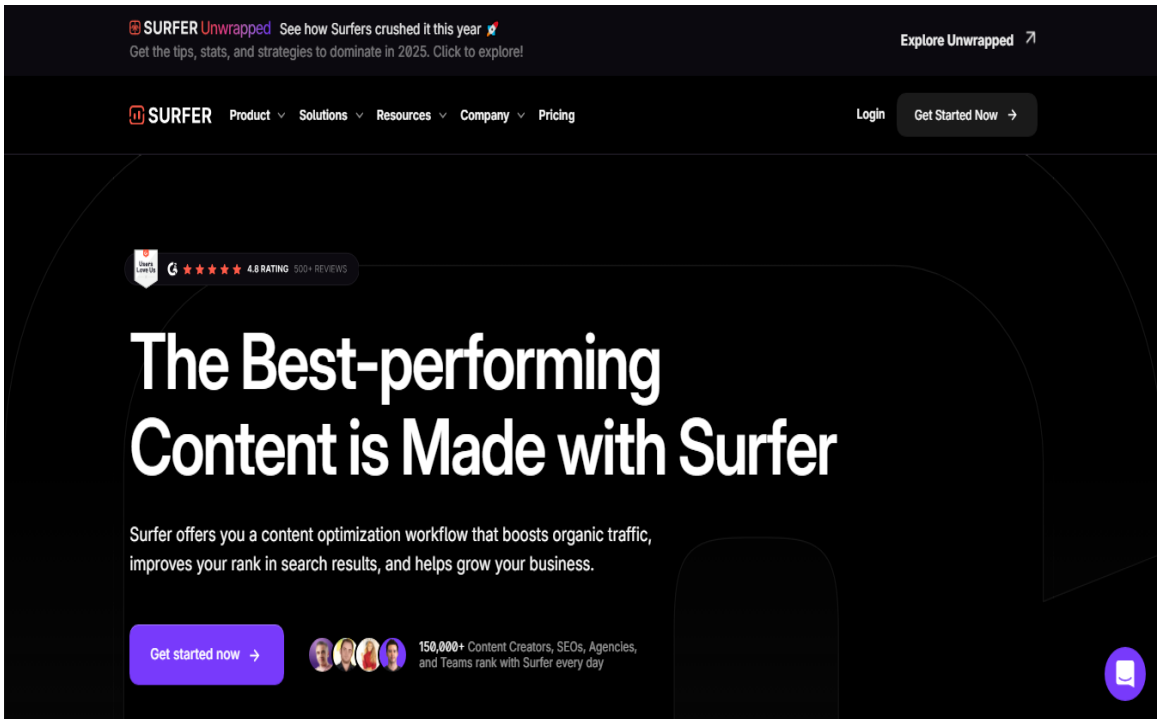
Another important consideration is security, because companies are wary about disclosing private online information. While preserving usability, using strong authentication procedures guarantees data security. Because digital products' success depends on their usefulness and convenience of use, a well-designed user interface is equally essential. Using an AI-powered SEO questionnaire to address these issues is in line with current research and technical trends, and it promises to improve security, accessibility, and efficiency while increasing sophisticated SEO analysis for companies of all sizes.

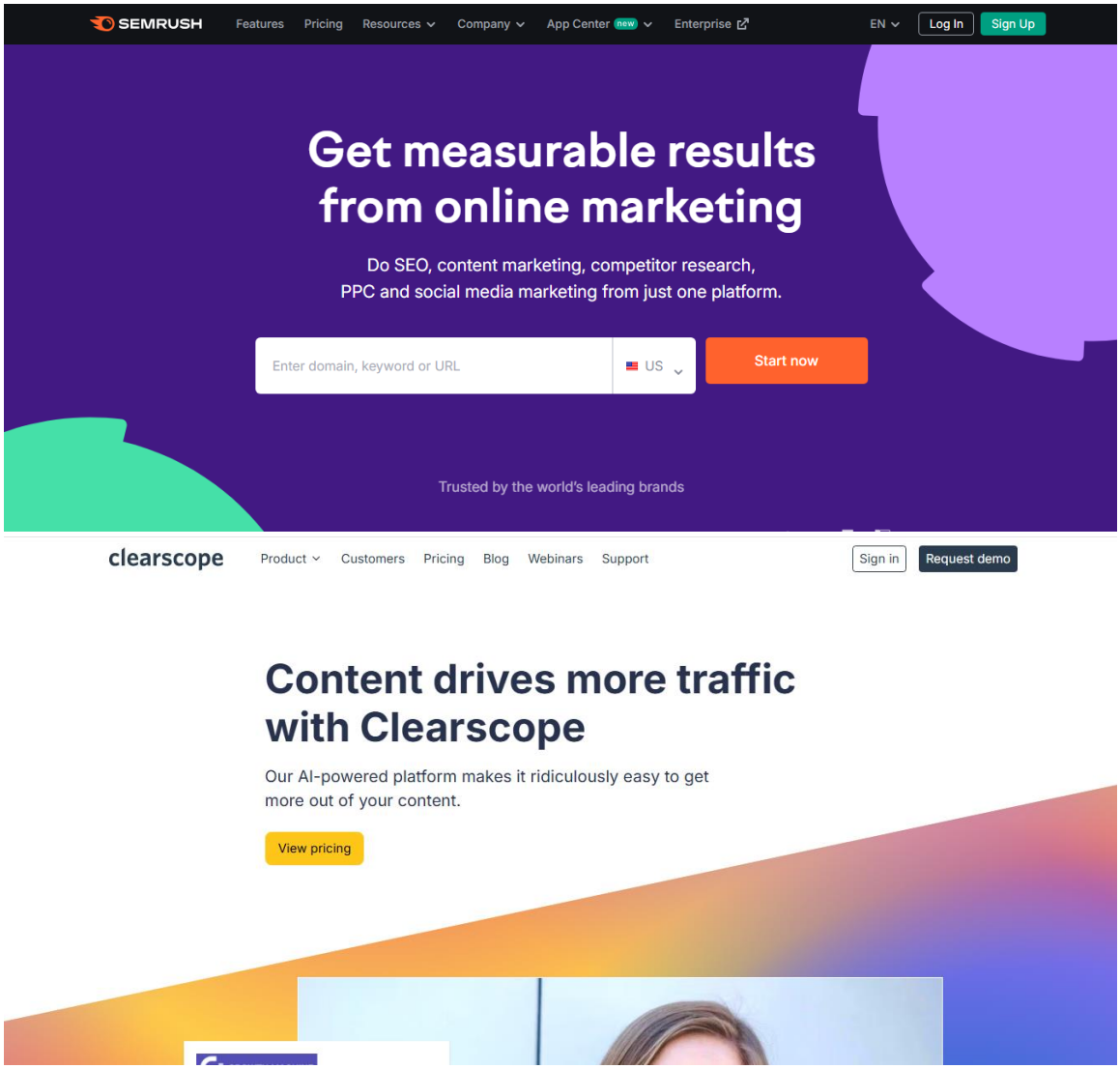
3.2 Discussion on problem solutions based on published articles

A number of issues with conventional SEO techniques are successfully addressed by the AI-powered SEO feedback system. By automating the analytical process, consistent, actionable results are ensured while requiring less manual labor. Businesses of all sizes

may benefit from the efficient SEO feedback generation made possible by AI models like GPT [1] [2]. Furthermore, the system's versatility in handling text, HTML, and API inputs guarantees wide accessibility and thorough analysis [3]. To protect sensitive data, security is given top priority through user authentication, encryption, and secure logins4.Usability is improved by the user-friendly design and static support pages like About Us and Contact Us[6].Last but not least, the platform's scalability enables it to expand with enterprises, overcoming the drawbacks of conventional tools[1] [7]. Businesses may enhance their SEO performance in a thorough, safe, and effective manner with these options.

3.3 Comparison of three/four leading solutions





Best Features:

- In this project, provide quick, precise feedback and automates the processing of analysis.
- Offers data-driven insights and optimization recommendations.
- Efficient and scalable, making it perfect for companies with big datasets.

Limitations

- Reliance on technology;
- Internet access fees;
- User acceptability and interface issues;

- Safety and secrecy concerns
- Might not have the sophisticated comprehension of a human specialist.
- Needs a learning curve to be used effectively.
- Less personalization than with manual analysis.

3.4 Recommended Approach

Table 1: Modules descriptions

Actuators	Functions
Client Side	● Sign In ● Clients can manually input text for SEO analysis. ● Users can upload HTML files, which will be converted into clean text for SEO analysis according to GPT-based AI model. ● If an API is provided, the system can extract data from the specified URL for SEO evaluation. ● The AI generates detailed SEO feedback in an easy-to-read report format.

CHAPTER 4

Methodology

4.1 What to Use

We'll create a system that generates SEO feedback reports using artificial intelligence. The project will support the following features: Client-side: Text contributions from customers are direct. If HTML data is provided, I will convert it to text. API Integration: If an API is available, I will get and handle the data. In response to pre-programmed system prompts, a chat GPT-based AI model will utilize the client's supplied text or HTML to produce SEO feedback. Along with the system architecture and theoretical underpinnings, the approach includes the procedure for identifying features pertinent to this project. It is acknowledged that the SDLC life cycle model, a comprehensive framework for design, development, etc., is the accepted method. I understand the many types of SDLC models. Software development employs a number of paradigms, including the Big Bang, Spiral, Waterfall, Agile, Iterative, and Dynamic system building models. Each of the models provides a framework to guide the development and use of the platform.

4.2 Why to use

The system architecture has to be defined as the initial stage in the development process. Determining the parts and how they interacted was a part of this. Scalability, security, and dependability were important components of the system infrastructure design. It required keeping the user interface and database administration separate from the product's back-end features. The design also included security measures to guarantee safe transactions and safeguard computer data. For every software project, the agile methodology must be applied. There are a lot of agile approach titles that I am aware with, like feature-driven creation, scrum, crystal, flexible system development technique, and kanban. Still, I used the DSDM technique to approach my project. The DSDM approach is advantageous for a number of reasons. Iterative development makes use of the dynamic system development approach, which allows for flexibility in changing needs. This tactic works well when prompt delivery is necessary. The Dynamic System Development Method requires software projects to be delivered on a regular basis. The company can implement

advantages sooner thanks to this agility strategy, which also enhances reporting on market changes and reduces time to market. To help team members see potential risks early in the project's lifespan, this agile approach incorporates risk management strategies into its operations. When precautions are implemented, risk management becoming less of an issue. For these reasons, I approached my project using the DSDM agile strategy.

4.3 Section of methodology

Pre-Project Phase:

- **Feasibility Study**: This stage entails assessing the technological, financial, and functional viability of the project proposal. Weighing the project's possible advantages, expenses, and risks is part of it.
- **Collecting condition**: During the conditions collection phase, the program's needs are gathered and recorded. It is crucial to comprehend company requirements, customer desires, and limitations while establishing the project's scope.
- **Planning:** Part of the planning process is developing an effort plan that details the project's goals, schedule, necessary resources, and deliverables. Determining the project's stakeholders, establishing roles and duties, and developing a plan for interaction and risk management are all essential.

Project Lifecycle Phase:

- Design: Using the requirements that have been acquired, the software design is constructed in this stage. Database, architectural, and user interface design are only a few of the high-level and comprehensive design tasks it involves.
- Development: Using the design requirements as a guide, the program is coded at this step. To produce a functional software product, the developer develops the source code, performs unit tests, and assembles components.
- Testing: By putting the program through its paces, this process aims to guarantee that it is both high-quality and functional. Unit, user validity, integration, and system testing are just a few of the many testing techniques it covers.

- Deployment: After undergoing a thorough testing process and approval, the software is placed into use. It is necessary to install, configure, and set up the program in the proper environment.

Post-Project Phase:

- Maintenance: The software moves into the maintenance stage following deployment. To make sure the program keeps functioning and adapts to changing requirements, this phase entails regular maintenance, bug repairs, and upgrades.
- Evaluation: By comparing the project's actual results with its stated goals, its efficacy may be assessed. It assists in determining what needs to be changed and what can be learned for future projects.
- Closure: The project officially ends at this moment. It include completing the project's documentation, monitoring its artifacts, and doing a project evaluation or post-mortem.

These parts offer a disciplined approach to managing software development projects and aid in achieving effective results, from early planning to post-deployment assistance**.**

4.4 Implementations plans

At this point in the project, the general public can now use the finished application. Once a bug has been identified and fixed, the new system must be enabled. This section selects the settings, protocols, and release requirements. The new system is then tested and put into use if everything works as planned.

Features:

- Text may be manually entered by clients for SEO analysis.
- The GPT-based AI model allows users to submit HTML files, which are then transformed into clean text for SEO analysis.
- The system may retrieve information from the given URL for SEO assessment if an API is supplied.
- The AI produces thorough SEO feedback in a report style that is simple to understand.

CHAPTER 5

Planning

5.1 Project Plan

The primary aim of the website is to provide the competitive digital world of today, improving online exposure and website performance need efficient search engine optimization, or SEO. In order to meet this demand, I suggest creating an AI-powered system that uses supplied website data to automatically generate SEO feedback. This cutting-edge technology will provide customers the option to contribute data via API integration, HTML file uploads, or manual entry. Superior GPT-based AI technology will be used by the system to evaluate the input and produce useful SEO insights. Beyond this, I may strive to achieve the following.

5.1.1 Management plan

Describe the project management procedure and the roles and duties of the project team. To guarantee a successful partnership, define the reporting and communication routes. Set up the issue-resolution approach's stages for decision-making and escalation.

Table 2: Management Planning

No	Task Name	Duration	Start Date	End Date
1	Introduction	10	1-7-24	10-7-24
2	Initial Study	5	10-7-24	15-7-24
3	Literature Review	5	16-7-24	20-7-24
4	Methodology	3	21-7-24	23-7-24
5	Planning	10	24-7-24	03-8-24
6	Feasibility	15	04-8-24	18-8-24
7	Foundation	5	19-8-24	23-8-24
8	Exploration	14	24-8-24	07-9-24
9	Engineering	30	08-9-24	08-10-24
10	Deployment	18	09-10-24	26-10-24
11	Testing	10	27-10-24	07-11-24

12	Implementation	5	08-11-24	12-11-24
13	Critical Appraisal and Evaluation	4	13-11-24	16-11-24
14	Lessons Learning	3	17-11-24	20-11-24
15	Conclusion	10	21-11-24	01-12-24
	Total	147 days		

5.1.2 Resource Allocation

Determine all of the project's resources, such as staff, tools, and software. Based on the workload and project schedule, decide how best to allocate the resources. Assign duties and obligations to team members while making sure they possess the requisite knowledge and abilities.

Table 3: Resource Allocation

No	Task Name	Duration	Resource
1	Introduction	10	End User
2	Initial Study	5	Analyst
3	Literature Review	5	Analyst
4	Methodology	3	Analyst
5	Planning	10	Analyst, Designer, Developer
6	Feasibility	15	Analyst
7	Foundation	5	Designer
8	Exploration	14	Designer , Developer
9	Engineering	30	Developer
10	Deployment	18	Analyst, Developer
11	Testing	10	Analyst, Developer, Tester, Users
12	Implementation	5	Analyst, Developer
13	Critical Appraisal and Evaluation	4	Analyst, Tester and Developer

14	Lessons Learning	3	Analyst, Users
15	Conclusion	10	Analyst
	Total	147 days	

5.1.3 Time Boxing

To make development and testing easier, divide the project into many time periods or iterations. Determine the duration of each time box as well as the tasks and outputs required for every iteration. For every time box, establish specific objectives and supply resources.

Table 4: Time Boxing

Time -Box	Task Name	Duration	Resource
TB1	Introduction	10	End Users, Analyst
	Initial Study	5	Analyst
	Literature Review	5	Analyst
TB2	Methodology	3	Analyst
	Planning	10	Analyst, Designer, Developer
	Feasibility	15	Analyst
TB3	Foundation	5	Designer
TB4	Exploration	14	Designer, Developer
	Engineering	30	Developer
TB5	Deployment	18	Analyst, Developer
	Testing	10	Analyst, Developer, Tester, Users
TB6	Implementation	5	Analyst, Developer
TB7	Critical Appraisal and Evaluation	4	Analyst, Tester and Developer
	Lessons Learning	3	Analyst, Users
TB8	Conclusion	10	Analyst

	Total	147 days	

CHAPTER 6

Feasibility

6.1 All possible types of feasibility

6.1.1 Operational feasibility

The possibility that all pertinent elements, such as engineering, planning, legal, and financial considerations, will be brought into account in order to effectively complete a project is ascertained by a feasibility study. Operational practicability is a measure of a system's ability to adapt to changing needs, leverage the scope defined during opportunity definition, would and satisfy project or requirement analysis criteria. Through the use of my online application, a traveler will be able to precisely and methodically manage every part of their journey in a timely manner. The web application at the heart of the suggested concept acts as a To address this need, I propose developing an AI-driven system that automatically generates SEO feedback based on provided website data. Customers will have the choice of contributing data through manual entry, HTML file uploads, or API connection thanks to this state-of-the-art technology.

6.1.2 Technical feasibility

Hardware	Software
Dell Laptop, Wi-Fi, Router, Cable, Android Phone	Android Studio, Google Chrome Browser, Windows, MS Word

6.1.3 Technology

Client slide	Server side
Html, css, JavaScript, React+Vite	Python

6.2 Cost Benefit Analysis

Cost-benefit analysis is a concept that project managers use to analyze the advantages and disadvantages of various project paths, including interactions, activities, business demands, and investments. The optimum way to achieve my goal at the lowest possible cost is determined via a cost-benefit analysis of all the choices available.

Project Name: RankMorph – AI Based SEO Analyzer.

Table 5: Cost Benefit

Equipment	1st Year	2nd Year	3rd Year	4th Year	Total
Web Based Application	20000				20000
Domain & Hosting		10000	10000	10000	30000
Software	1000				1000
Internet	2000	2000	2000	2000	8000
Training	5000				5000
Development		5000			5000
Maintenance	10000	10000	10000	10000	4000
Total					73,000 BDT.

6.3 DSDM Dynamic System Development Method (DSDM)

Instead, then focusing on a particular application development tool or technology, the Dynamic Systems Development Method, or DSDM for short, is an agile project and software development management organizational structure. It strongly emphasizes iterative development methods, regular software delivery, and cooperation between

development teams and business stakeholders. It is vital to remember that DSDM does not need the use of certain tools or technologies, such as. These web development tools are widely known and definitely useful for DSDM applications.

CHAPTER 7

Foundation

7.1 Some potential approaches

7.1.1 Interview

It is crucial to set up interviews for every project. By conducting interviews, Multiple input sources, such as human text, HTML files, and API interface, are used to automate SEO analysis in this AI-powered SEO feedback system, which has been described as a flexible tool. It effectively analyzes data and uses GPT-based AI to produce personalized reports with meaningful feedback. In order to improve the user experience, the interview emphasized its secure access features, user-friendly design, and static sites such About Us and Contact Us. The focus was on how it is appropriate for companies of all sizes because to its scalability, flexibility, and strong security features like access control and encryption. Businesses may efficiently improve their online presence thanks to ongoing upgrades that guarantee relevancy with changing SEO trends.

7.1.2 Observation

An effective method for expediting SEO analysis and feedback production is the AI-powered SEO feedback system. Accessibility and usability for a wide range of customers are guaranteed by its multi-format handling capabilities, secure access, and intuitive design. Its efficiency is demonstrated by the incorporation of GPT-based AI for actionable insights, and its flexibility in responding to evolving SEO trends and business expansion is demonstrated by features like scalability and ongoing upgrades. All things considered, the system provides a safe, adaptable, and efficient platform for raising internet presence.

7.2 Specific problem are identification and description

The ineffectiveness and irregularity of conventional SEO analysis techniques are the main issues that the AI-powered SEO feedback system attempts to solve. For small and medium-sized companies with limited resources, manual procedures are impracticable since they can demand a great deal of time, effort, and skill. Furthermore, accessibility and scalability are limited by the absence of automated tools for managing various data types, including

text, HTML, and API inputs. The procedure is made more difficult by security issues, such as illegal access to private website data. These difficulties are made worse by the ever-changing landscape of SEO trends, which necessitate regular modifications and adjustments that are difficult to achieve using conventional approaches. This makes it evident that a flexible, automated, and trustworthy solution is required to improve website optimization efforts and expedite SEO feedback.

7.3 Possible solution

The AI-powered SEO input system automates the SEO analysis process, providing a creative solution to the issues found. By using GPT-based AI, the system successfully generates actionable SEO feedback, doing away with the laborious and irregular nature of human procedures. Text, HTML, and API are just a few of the input types it accepts, guaranteeing accessibility and adaptability for a range of customer requirements. Sensitive information is protected thanks to user authentication, data encryption, and secure login mechanisms, which solve important security issues. The platform can handle big datasets and expand to meet corporate demands because to its scalability, and it is updated often to stay relevant with changing SEO trends. Businesses may efficiently enhance their internet presence with this all-inclusive strategy, which offers a user-friendly, safe, and flexible solution.

7.4 Overall Requirement List

- Functional Requirements
- Non-Functional Requirements.

7.4.1 Functional Requirements

7.4.1.1 Client side

- Sign In
- Clients can manually input text for SEO analysis.
- Users can upload HTML files, which will be converted into clean text for SEO analysis according to GPT-based AI model.
- If an API is provided, the system can extract data from the specified URL for SEO evaluation.

- The AI generates detailed SEO feedback in an easy-to-read report format.

7.4.2 Non-Functional Requirements

7.4.2.1 Security

Each user of the system has an account, and only those who have been given authorization and a password may access it. I use both JavaScript to encrypt the credentials.

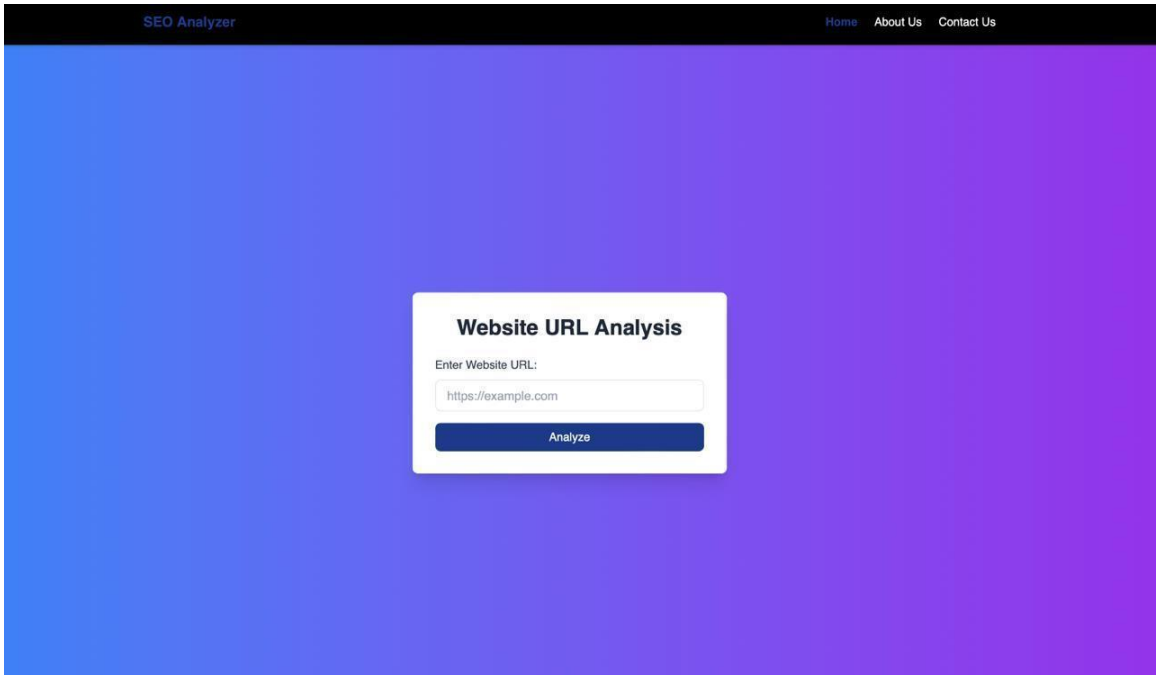


Fig 1: User Interfaces

7.4.2.2 Performance

Records are simple to keep up to date.

7.4.2.3 Availability

Users merely need a PC with an Internet connection to access the system from anywhere at any time. Numerous web browsers, including as Internet Explorer, Mozilla, Opera, and Chrome, are compatible with the system.

7.4.2.4 User Friendly

It has an engaging user interface and is easy to use.

- The website shouldn't have any noticeable latency or outages when many people are accessing it at once.
- Managing large amounts of data is a must for the website.
- The website should have a straightforward, user-friendly design and layout.
- The website must easily accommodate new features and functionalities without requiring extensive editing or redesigning.
- To fix defects and issues that arise after launch, the website must be maintained.

7.5 Which technology to be implemented

The program I'm working on is entirely web based. I'm developing my project with html, css, java script, python.

HTML: Web pages can be created using markup languages like HTML. HTML files may be seen and understood by web browsers. Every website is built on HTML components. enables you to create engaging content by permitting the usage of HTML elements and visuals. In [8], it may also produce chapters, lists, quotes, titles, and links.

CSS: To further personalize the material on my webpage, I may utilize CSS to alter the fonts, colors, and layouts. This improves the cohesion and aesthetic attractiveness of my website. The website seems friendlier as a consequence. Inside [9]

JavaScript: JavaScript is one of the most popular programming languages available today. I also utilize JavaScript as a language for web development. In doing so, a layer of common web technologies is developed. Within [10]

Bootstrap: Version 4.0 of Bootstrap is currently in use. All the HTML, CSS, and JavaScript components you need to develop responsive websites can be found in Bootstrap 4. As a result, I created user accounts on my website.[11]

7.6 Recommendation and justifications

To overcome the shortcomings and inefficiencies of conventional SEO techniques, it is advised to put the AI-powered SEO feedback system into place. By automating the analytical process with GPT-based AI, this solution eliminates the need for a significant amount of human labor while producing faster and more reliable feedback. Its ability to accept text, HTML, and API as input formats guarantees adaptability and meets the various demands of companies. By including strong security measures like user identification and data encryption, the dangers of unwanted access to private data are reduced. Additionally, because of the system's scalability, companies of all sizes may use it to handle their expanding data needs. Long-term relevance is ensured by regular changes in line with changing SEO trends, giving firms a competitive advantage. The system is a useful instrument for improving online visibility and attaining sustained digital growth because of its efficiency, flexibility, security, and adaptability.

CHAPTER 8

Exploration

8.1 Use case

This section analyzes both functional and non-functional needs using use-case data and visuals.

Client slide:

- Sign In
- Clients can manually input text for SEO analysis.
- Users can upload HTML files, which will be converted into clean text for SEO analysis according to GPT-based AI model.
- If an API is provided, the system can extract data from the specified URL for SEO evaluation.
- The AI generates detailed SEO feedback in an easy-to-read report format.

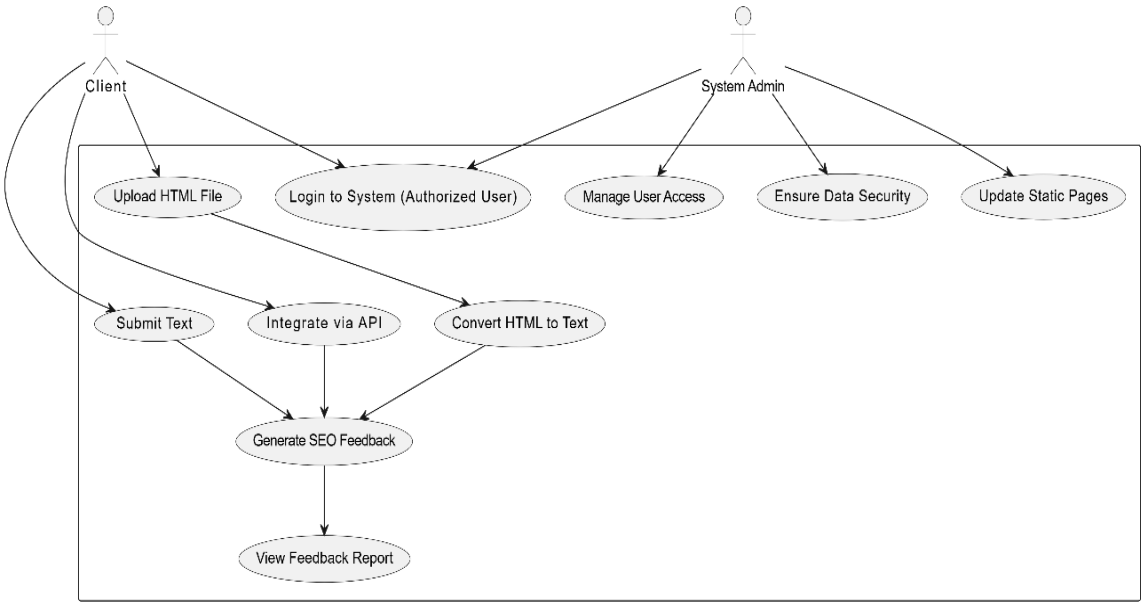


Fig 3: Use case Diagram

8.2 Activity diagram

Describe how the system is dynamic. It looks like a flow chart that shows how one task interrelates with another. You may use the exercise to describe how the system works. Consequently, control is distributed across all operations. The following is the complete activity diagram for each module: responsible.

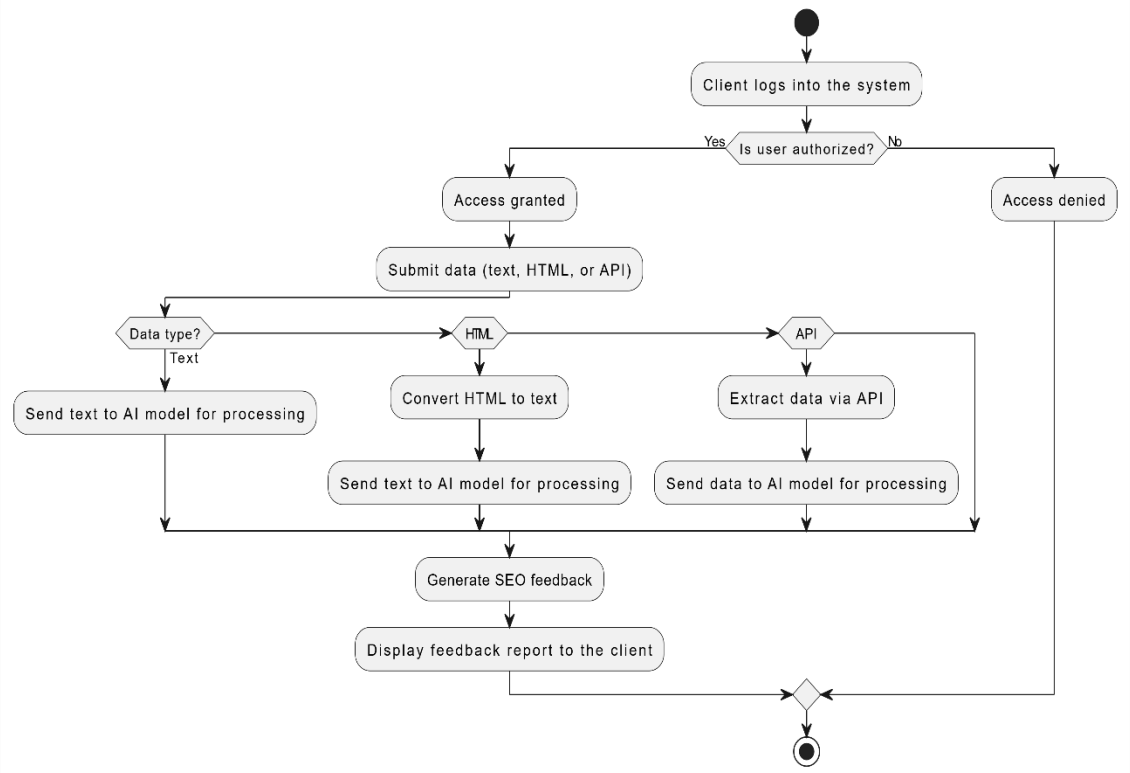


Fig 4: Activity Diagram.

8.3 Requirement catalogue

Functional requirements:

- FR1: Sign In if authorized account.
- FR2: Clients can manually input text for SEO analysis.
- FR3: Users can upload HTML files, which will be converted into clean text for SEO analysis according to GPT-based AI model.
- FR5: If an API is provided, the system can extract data from the specified URL for SEO evaluation.
- FR6: The AI generates detailed SEO feedback in an easy-to-read report format.

Non-Functional Requirements:

- NFR1: Each user of the system is given an account, and only those who have been granted authorization and a password can access it. I use both JavaScript to encrypt the credentials.
- NFR2: Maintaining and updating records is simple.
- NFR3: Users simply need a PC with an Internet connection to access the system from anywhere at any time. Numerous web browsers, including as Internet Explorer, Mozilla, Opera, and Chrome, are compatible with the system.
- NFR4: The technology has an engaging user interface and is easy to use.

User Interface Requirements:

- UIR1: A user-friendly interface with intuitive navigation that makes accessing various functions and capabilities simple.
- UIR2: responsive design can adapt to different screen and device sizes.
- UIR3: Icons offer visual signals to help users comprehend and operate the system.

Security and Privacy Requirements:

- SR1: Safe authentication and authorization procedures to safeguard user information and accounts.
- SR2: Protect all data from various user accounts, web application data.

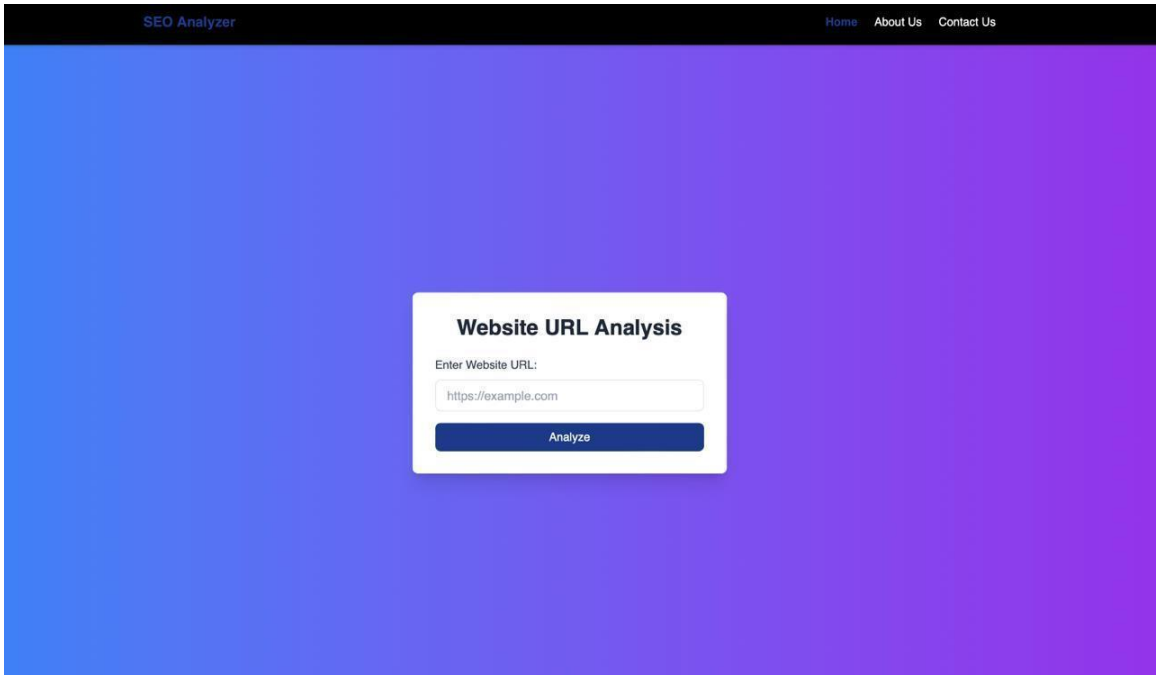
8.4 Prioritized Requirement List (PRL)

Table 6: Prioritized requirement list

Requirement ID	Requirement Description	Priority	Dependencies	Status	Validation Criteria
RQ1	Only Authorized person can login	High			successfully done
RQ2	Login functionality	High	RQ1		Client can sign in successfully.

RQ3	Clients can manually input text for SEO analysis.	High	RQ2		successfully
RQ4	Users can upload HTML files, which will be converted into clean text for SEO analysis according to GPT-based AI model.	High	RQ2, RQ3		successfully
RQ5	If an API is provided, the system can extract data from the specified URL for SEO evaluation.	High	RQ2, RQ3		successfully

8.5 Prototype of new system



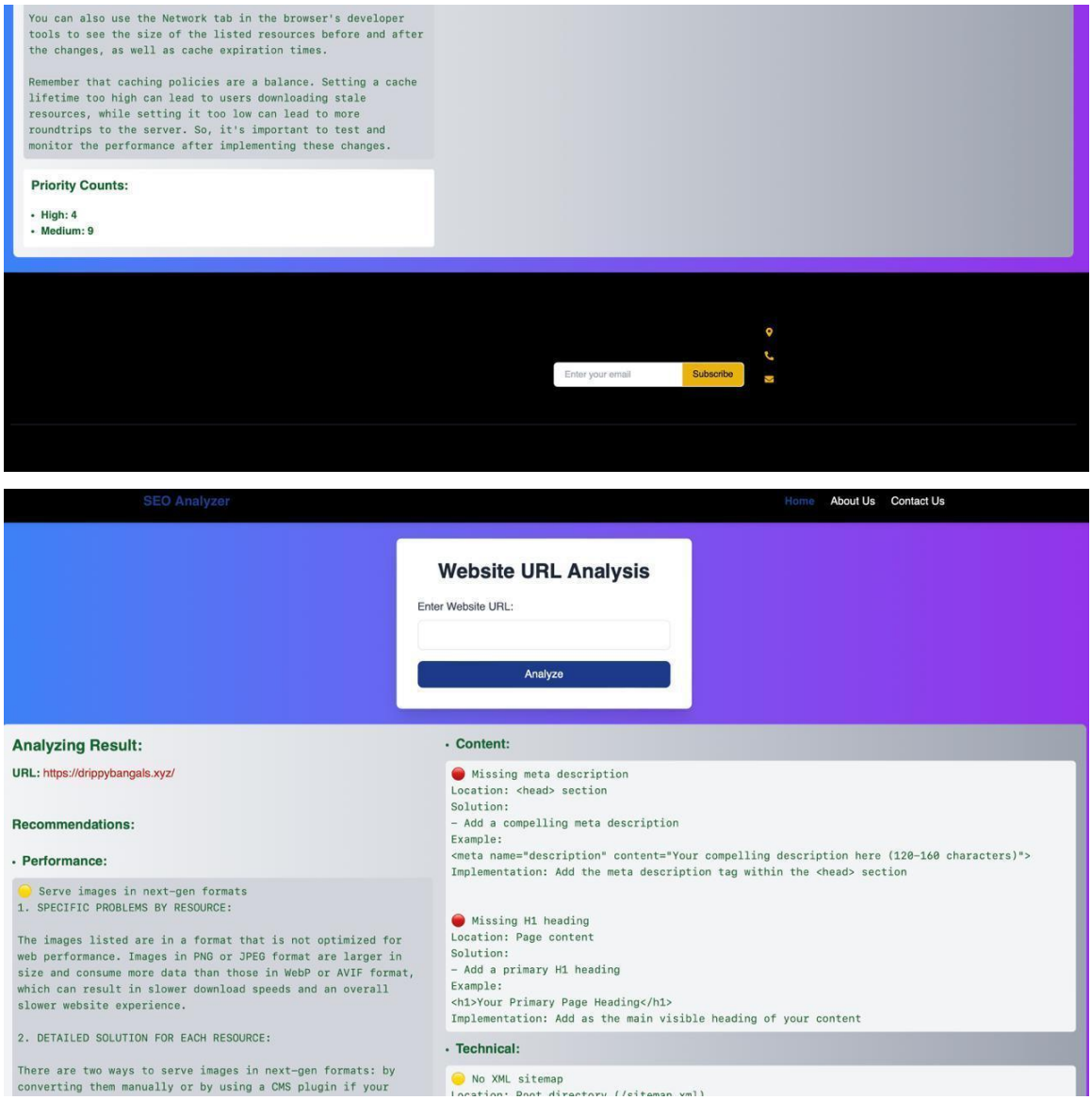


Fig 5: Interfaces for client side

CHAPTER 9

Engineering

Today's competitive digital world necessitates effective search engine optimization, or SEO, to improve online visibility and website performance. This is the website's main goal. I propose developing an AI-powered system that automatically generates SEO feedback based on the website data in order to satisfy this need. Customers will be able to submit data using this state-of-the-art technology through manual entry, HTML file uploads, or API connectivity. The system will analyze the input using advanced GPT-based AI technology to generate valuable SEO insights. Beyond this, I may aim for the following.

Features:

- Sign In if authorized account.
- Clients can manually input text for SEO analysis.
- Users can upload HTML files, which will be converted into clean text for SEO analysis according to GPT-based AI model.
- If an API is provided, the system can extract data from the specified URL for SEO evaluation.
- The AI generates detailed SEO feedback in an easy-to-read report format.

9.1 Class Diagram

The source material for the interclass linkages was displayed in a class. The class in this instance defines the variables and actions of an entity, either as a single programming description or as a distinct entity inside a program.

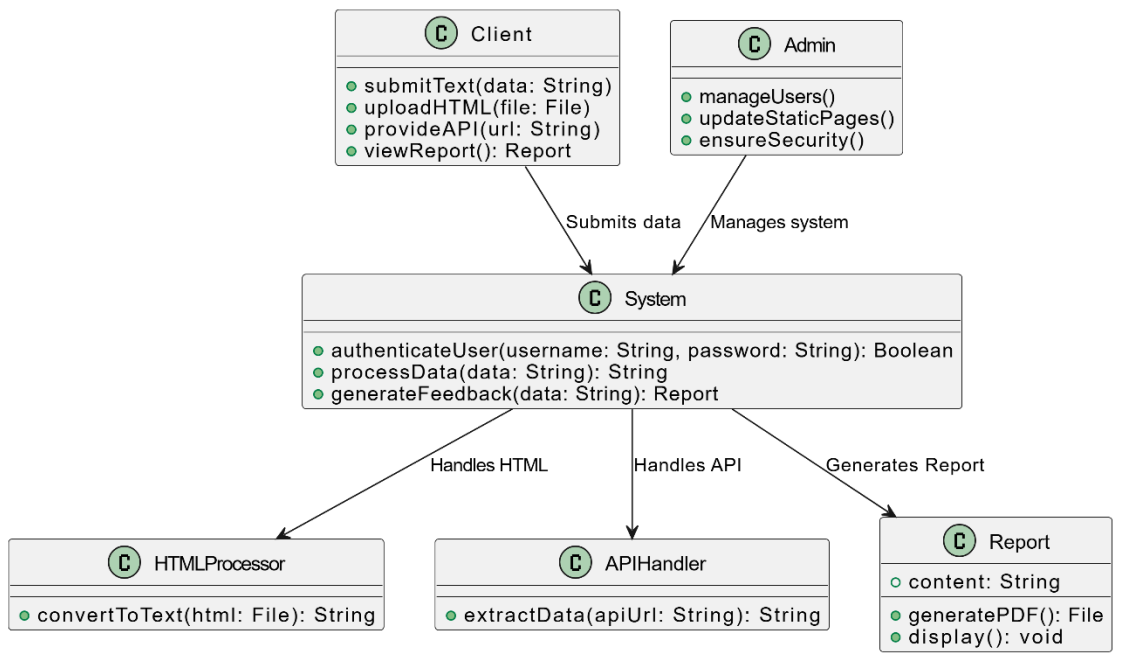


Fig 6: Class Diagram

9.2 ER diagram

In design, institutional means of communication—also known as the ER model, ER Diagram, or ERD—are a kind of structural application. The key components of the restricted system and the relationships among these entities are the only significant pieces of information that the ERD exchanges and displays in a different way. This completes the client module's entity-relationship diagram.

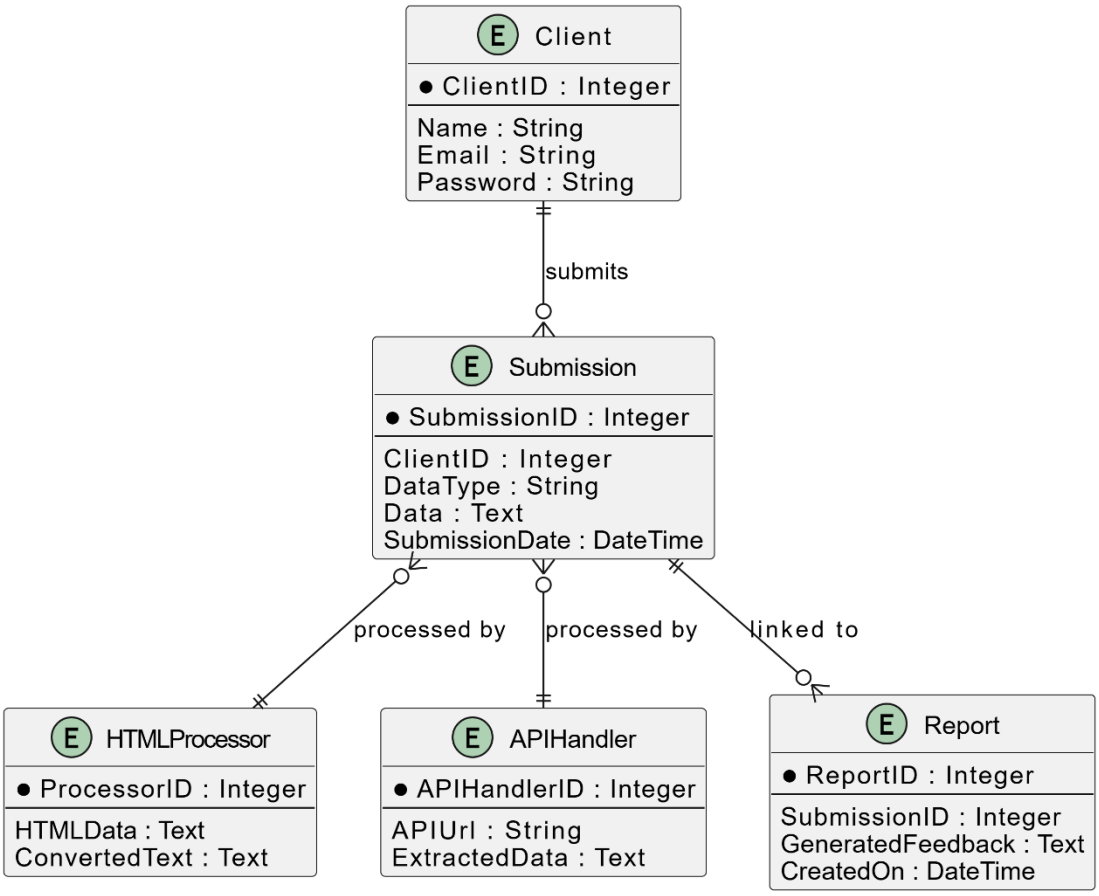


Fig 8: ER Diagram.

9.3 Sequence Diagram

This is all about one module's sequence diagram: Client side

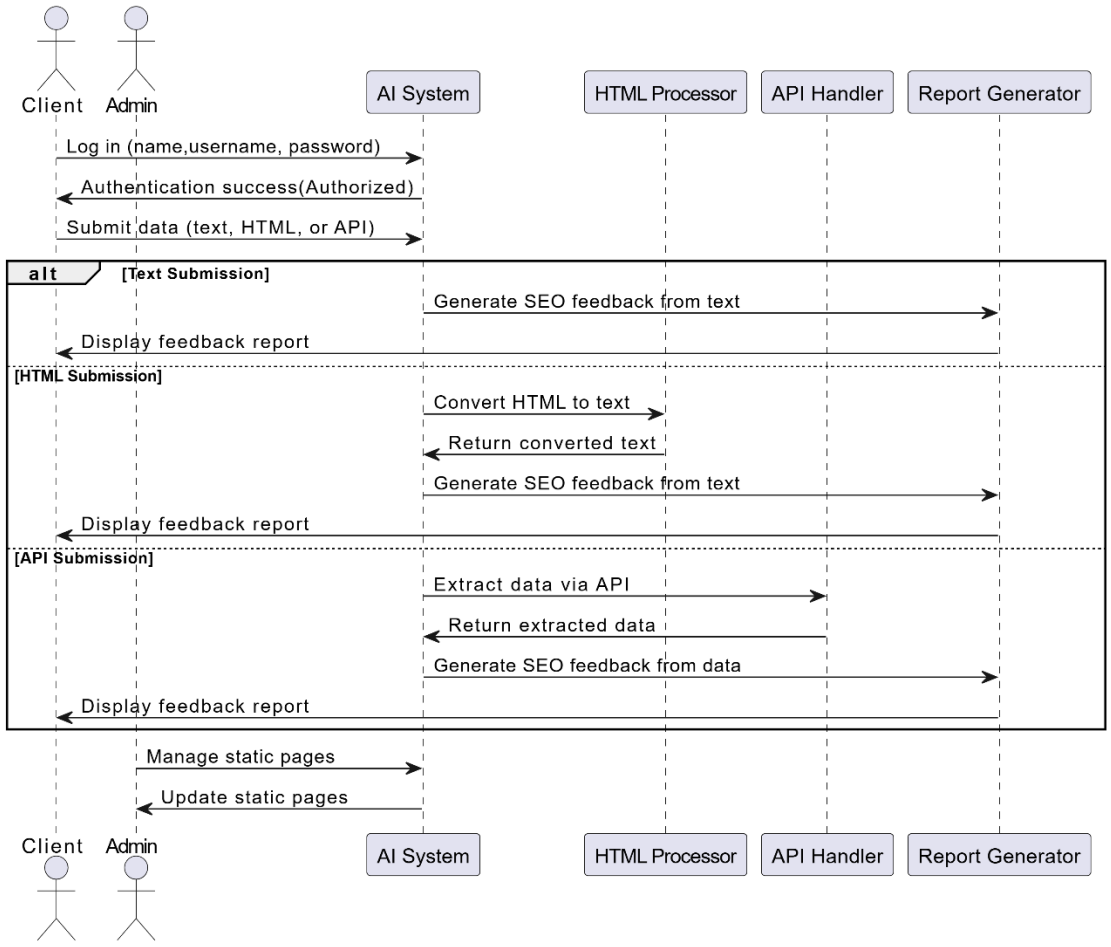


Fig 9: Sequence Diagram

CHAPTER 10

Development

10.1 Core Module Samples

```python
from fastapi import FastAPI, HTTPException
from pydantic import BaseModel, HttpUrl
from app.modules import pagespeed, content_analysis, technical_seo, recommendations
from app.logger import logger
from fastapi.middleware.cors import CORSMiddleware

app = FastAPI()

class URLInput(BaseModel):
    url: HttpUrl

app.add_middleware(
    CORSMiddleware,
    allow_origins=["*"],  # Allows all origins
    allow_credentials=True,
    allow_methods=["*"],  # Allows all HTTP methods
    allow_headers=["*"],  # Allows all headers
)


@app.post("/analyze")
async def analyze_url(input: URLInput):
    url = str(input.url)
    logger.info(f"Analyzing URL: {url}")
```

```python

@app.post("/analyze")
async def analyze_url(input: URLInput):
    url = str(input.url)
    logger.info(f"Analyzing URL: {url}")

    try:
        pagespeed_insights = pagespeed.get_insights(url)
        content_data = content_analysis.analyze(url)
        tech_seo_data = technical_seo.analyze(url)
        recommendations_data = recommendations.generate(url, pagespeed_insights, content_data, tech_seo_data)

        return {
            "url": url,

            "recommendations": recommendations_data

        }
    except Exception as e:
        logger.error(f"Error analyzing URL: {str(e)}")
        raise HTTPException(status_code=500, detail=str(e))

```

```python
import requests
from fastapi import HTTPException
from ..config import PAGESPEED_API_KEY
from ..logger import logger

def get_insights(url):
    logger.info(f"Fetching PageSpeed Insights for {url}")
    api_url = f"https://www.googleapis.com/pagespeedonline/v5/runPagespeed?url={url}&key={PAGESPEED_API_KEY}"
    response = requests.get(api_url)
    if response.status_code != 200:
        logger.error(f"Failed to fetch PageSpeed Insights: {response.status_code}")
        raise HTTPException(status_code=400, detail="Failed to fetch PageSpeed Insights")
    data = response.json()
    return {
        'performance_score': data['lighthouseResult']['categories']['performance']['score'],
        'opportunities': data['lighthouseResult']['audits']
    }
```

```python
import requests
from bs4 import BeautifulSoup
from ..logger import logger

def analyze(url):
    logger.info(f"Analyzing content for {url}")
    response = requests.get(url)
    soup = BeautifulSoup(response.text, 'html.parser')

    title = soup.title.string if soup.title else None
    meta_description = soup.find('meta', attrs={'name': 'description'})
    meta_description = meta_description['content'] if meta_description else None

    h1_tags = [h1.text for h1 in soup.find_all('h1')]
    img_alt_texts = [img.get('alt', '') for img in soup.find_all('img')]

    content = ' '.join([p.text for p in soup.find_all('p')])
    word_count = len(content.split())

    return {
        'title': title,
        'meta_description': meta_description,
        'h1_tags': h1_tags,
        'img_alt_texts': img_alt_texts,
        'word_count': word_count
    }
```

```python
import requests
from urllib.parse import urlparse
from ..logger import logger

def analyze(url):
    logger.info(f"Analyzing technical SEO for {url}")
    parsed_url = urlparse(url)
    is_https = parsed_url.scheme == 'https'

    response = requests.get(url)
    headers = response.headers

    has_robots_txt = requests.get(f"{parsed_url.scheme}://{parsed_url.netloc}/robots.txt").status_code == 200
    has_sitemap = requests.get(f"{parsed_url.scheme}://{parsed_url.netloc}/sitemap.xml").status_code == 200

    return {
        'is_https': is_https,
        'server': headers.get('Server'),
        'has_robots_txt': has_robots_txt,
        'has_sitemap': has_sitemap,
        'status_code': response.status_code
    }
```

```python
import tiktoken
from openai import OpenAI
from typing import Dict, Any, List, Optional
from ..config import OPENAI_API_KEY
from ..logger import logger

client = OpenAI(api_key=OPENAI_API_KEY)

# Keep your existing utility functions
def count_tokens(text: str) -> int:
    try:
        encoding = tiktoken.encoding_for_model("gpt-4")
        return len(encoding.encode(text))
    except Exception as e:
        logger.error(f"Error counting tokens: {e}")
        return len(text.split())

def truncate_text(text: str, max_tokens: int = 1000) -> str:
    if not text:
        return ""
    encoding = tiktoken.encoding_for_model("gpt-4")
    tokens = encoding.encode(text)
    if len(tokens) > max_tokens:
        return encoding.decode(tokens[:max_tokens]) + "..."
    return text
```

```python
# Add new class for enhanced issue tracking
class PerformanceIssue:
    def __init__(self,
                 issue_type: str,
                 title: str,
                 score: float,
                 description: str,
                 impact: Dict[str, Any],
                 location: Optional[Dict[str, Any]] = None,
                 resources: List[Dict[str, Any]] = None):
        self.issue_type = issue_type
        self.title = title
        self.score = score
        self.description = description
        self.impact = impact
        self.location = location or {}
        self.resources = resources or []

    def get_severity(self) -> str:
        if self.score < 0.5:
            return 'critical'
        elif self.score < 0.9:
            return 'warning'
        return 'info'
```

```python
    def to_dict(self) -> Dict[str, Any]:
        return {
            'type': self.issue_type,
            'title': self.title,
            'severity': self.get_severity(),
            'score': self.score,
            'description': self.description,
            'location': self.location,
            'impact': self.impact,
            'resources': self.resources
        }

# Keep and enhance your location extraction function
def extract_resource_location(item: dict) -> dict:
    """Extract specific location information from a resource"""
    location = {
        'file': '',
        'line': None,
        'column': None,
        'selector': '',
        'context': '',
        'url': item.get('url', '')  # Add URL to location info
    }
```

```python
# Keep and enhance your location extraction function
def extract_resource_location(item: dict) -> dict:
    """Extract specific location information from a resource"""
    location = {
        'file': '',
        'line': None,
        'column': None,
        'selector': '',
        'context': '',
        'url': item.get('url', '')  # Add URL to location info
    }

    # Extract source location
    if 'source' in item:
        location.update({
            'file': item['source'].get('file', ''),
            'line': item['source'].get('line', None),
            'column': item['source'].get('column', None)
        })
```

```python
    # Extract DOM location for elements
    if 'node' in item:
        node = item['node']
        location.update({
            'selector': node.get('selector', ''),
            'context': node.get('snippet', ''),
            'nodeLabel': node.get('nodeLabel', ''),
            'path': node.get('path', ''),
            'boundingRect': node.get('boundingRect', {})
        })

    return location

# Keep your format functions with minor enhancements
def format_location(location: dict) -> str:
    """Format location information for display"""
    parts = []

    if location['file']:
        parts.append(f"File: {location['file']}")
        if location['line']:
            parts.append(f"Line: {location['line']}")
        if location['column']:
            parts.append(f"Column: {location['column']}")
```

```python
# Keep your format functions with minor enhancements
def format_location(location: dict) -> str:
    """Format location information for display"""
    parts = []

    if location['file']:
        parts.append(f"File: {location['file']}")
        if location['line']:
            parts.append(f"Line: {location['line']}")
        if location['column']:
            parts.append(f"Column: {location['column']}")

    if location['selector']:
        parts.append(f"DOM: {location['selector']}")
        if location.get('nodeLabel'):
            parts.append(f"Element: {location['nodeLabel']}")

    if location.get('url'):
        parts.append(f"URL: {location['url']}")

    return ', '.join(parts) if parts else "Location not available"

def format_size(bytes: int) -> str:
    """Format byte sizes in human-readable format"""
    for unit in ['B', 'KB', 'MB', 'GB']:
        if bytes < 1024:
            return f"{bytes:.1f}{unit}"
        bytes /= 1024
    return f"{bytes:.1f}TB"
```

```python
def format_size(bytes: int) -> str:
    """Format byte sizes in human-readable format"""
    for unit in ['B', 'KB', 'MB', 'GB']:
        if bytes < 1024:
            return f"{bytes:.1f}{unit}"
        bytes /= 1024
    return f"{bytes:.1f}TB"

# Enhance your extract_performance_data function
def extract_performance_data(pagespeed_data: dict) -> dict:
    """Extract and categorize comprehensive performance data with enhanced issue tracking"""
    # Initialize with categories for different types of audits
    results = {
        'opportunities': [],  # Direct optimization opportunities
        'diagnostics': [],    # Technical diagnostics
        'metrics': [],        # Core Web Vitals and metrics
        'animations': [],     # Animation-related issues
        'scripting': [],      # JavaScript execution issues
        'network': [],        # Network and server issues
        'layout': []         # Layout and visual stability issues
    }
    results = {
        'opportunities': [],
        'diagnostics': [],
        'metrics': []
    }
```

```python
    try:
        # Define comprehensive audit categories with patterns
        categories = {
            'loading': ['largest-contentful-paint', 'first-contentful-paint', 'speed-index', 'server-response-time'],
            'interactivity': ['interactive', 'blocking-time', 'cpu-time', 'bootup-time', 'mainthread-work-breakdown'],
            'visual_stability': ['cumulative-layout-shift', 'layout-shift-elements', 'non-composited-animations'],
            'resources': ['render-blocking-resources', 'unused-javascript', 'unused-css', 'uses-text-compression'],
            'images': ['modern-image-formats', 'sized-images', 'responsive-images', 'offscreen-images', 'uses-optimized-images'],
            'caching': ['uses-long-cache-ttl', 'efficient-cache-policy'],
            'timing': ['total-blocking-time', 'time-to-first-byte', 'max-potential-fid'],
            'script_optimization': ['third-party-facades', 'third-party-summary', 'legacy-javascript'],
            'dom': ['dom-size', 'duplicated-javascript', 'legacy-javascript', 'unsized-images'],
            'network': ['network-requests', 'network-rtt', 'network-server-latency']
        }

        for audit_id, audit in pagespeed_data.items():
            if not isinstance(audit, dict) or audit.get('score') is None:
                continue

            score = float(audit.get('score', 1))
            if score >= 0.9:  # Skip high-scoring audits
                continue

            # Create enhanced issue object
            issue = PerformanceIssue(
                issue_type=determine_category(audit_id, categories),
                title=audit.get('title', ''),
```

```
                # Create enhanced issue object
                issue = PerformanceIssue(
                    issue_type=determine_category(audit_id, categories),
                    title=audit.get('title', ''),
                    score=score,
                    description=audit.get('description', ''),
                    impact={
                        'displayValue': audit.get('displayValue', ''),
                        'numericValue': audit.get('numericValue'),
                        'numericUnit': audit.get('numericUnit', '')
                    }
                )

                # Extract resource details with enhanced location tracking
                if 'details' in audit and 'items' in audit['details']:
                    for item in audit['details']['items'][:5]:  # Limit to top 5 items
                        resource = {
                            'url': item.get('url', ''),
                            'type': item.get('resourceType', ''),
                            'size': item.get('totalBytes', 0),
                            'wasted': item.get('wastedBytes', 0),
                            'time': item.get('wastedMs', 0),
                            'location': extract_resource_location(item)
                        }
                        issue.resources.append(resource)

                # Categorize based on audit type and characteristics
```

```
                # Categorize based on audit type and characteristics
                audit_type = determine_audit_type(audit)

                # Extract specialized metrics based on audit type
                if audit_type == 'animation':
                    enhance_animation_data(issue, audit)
                elif audit_type == 'scripting':
                    enhance_script_data(issue, audit)
                elif audit_type == 'layout':
                    enhance_layout_data(issue, audit)

                # Add to appropriate category
                if audit_type in results:
                    results[audit_type].append(issue)
                elif audit.get('details', {}).get('type') == 'opportunity':
                    results['opportunities'].append(issue)
                else:
                    results['diagnostics'].append(issue)

        except Exception as e:
            logger.error(f"Error extracting performance data: {e}")

        return results
```

```python
def determine_audit_type(audit: Dict[str, Any]) -> str:
    """Determine the specific type of audit"""
    audit_id = audit.get('id', '').lower()

    if 'animation' in audit_id or audit_id == 'non-composited-animations':
        return 'animations'
    elif any(term in audit_id for term in ['script', 'javascript', 'bootup-time']):
        return 'scripting'
    elif any(term in audit_id for term in ['layout-shift', 'cumulative-layout-shift']):
        return 'layout'
    elif any(term in audit_id for term in ['server', 'network', 'rtt']):
        return 'network'
    elif audit.get('details', {}).get('type') == 'opportunity':
        return 'opportunities'

    return 'diagnostics'

def enhance_animation_data(issue: PerformanceIssue, audit: Dict[str, Any]) -> None:
    """Add animation-specific data to the issue"""
    if 'details' in audit and 'items' in audit['details']:
        for item in audit['details']['items']:
            if 'node' in item:
                # Extract animation properties
                animation_info = {
                    'element': item['node'].get('selector', ''),
                    'snippet': item['node'].get('snippet', ''),
                    'properties': []
                }
```

```python
                # Extract failing animation properties
                if 'subItems' in item and 'items' in item['subItems']:
                    for subitem in item['subItems']['items']:
                        if 'failureReason' in subitem and 'animation' in subitem:
                            animation_info['properties'].append({
                                'property': subitem['animation'],
                                'reason': subitem['failureReason']
                            })

                issue.resources.append({
                    'type': 'animation',
                    'details': animation_info
                })

def enhance_script_data(issue: PerformanceIssue, audit: Dict[str, Any]) -> None:
    """Add script-specific data to the issue"""
    if 'details' in audit and 'items' in audit['details']:
        for item in audit['details']['items']:
            script_info = {
                'url': item.get('url', ''),
                'total_cpu_time': item.get('total', 0),
                'script_evaluation_time': item.get('scripting', 0),
                'parsing_time': item.get('scriptParseCompile', 0)
            }

            issue.resources.append({
                'type': 'script',
                'details': script_info
```

```python
def enhance_layout_data(issue: PerformanceIssue, audit: Dict[str, Any]) -> None:
    """Add layout-specific data to the issue"""
    if 'details' in audit:
        # Extract CLS contributors
        if 'items' in audit['details']:
            for item in audit['details']['items']:
                layout_info = {
                    'shift_score': item.get('score', 0),
                    'element': item.get('node', {}).get('selector', ''),
                    'snippet': item.get('node', {}).get('snippet', ''),
                    'timing': {
                        'startTime': item.get('startTime', 0),
                        'duration': item.get('duration', 0)
                    }
                }

                issue.resources.append({
                    'type': 'layout',
                    'details': layout_info
                })


def determine_category(audit_id: str, categories: Dict[str, List[str]]) -> str:
    """Determine the category of a performance issue"""
    audit_id_lower = audit_id.lower()
    for category, patterns in categories.items():
        if any(pattern in audit_id_lower for pattern in patterns):
            return category
```

```python
# Enhance your get_performance_recommendations function
def get_performance_recommendations(issue: PerformanceIssue) -> str:
    """Generate detailed recommendations for a performance issue"""
    try:
        # Prepare detailed resource analysis
        resource_analysis = []
        for resource in issue.resources:
            location = format_location(resource['location'])

            # Extract DOM details if available
            dom_details = ""
            if resource['location'].get('selector'):
                dom_details = f"""
DOM Location:
- Selector: {resource['location']['selector']}
- Element: {resource['location'].get('nodeLabel', 'Not specified')}
- HTML Context: {resource['location'].get('context', 'Not available')}"""

            # Extract file details if available
            file_details = ""
            if resource['location'].get('file'):
                file_details = f"""
File Location:
- Path: {resource['location']['file']}
- Line: {resource['location'].get('line', 'Not specified')}
- Column: {resource['location'].get('column', 'Not specified')}"""

```

```
            # Format resource analysis with detailed breakdown
            analysis = f"""
Resource Details:
URL: {resource['url']}
Type: {resource.get('type', 'Not specified')}
Current Size: {format_size(resource['size'])}
Potential Savings: {format_size(resource['wasted'])}
Time Impact: {resource['time']}ms
{dom_details}
{file_details}
Current Implementation Issues:
- Size: {format_size(resource['size'])} (Can be optimized to save {format_size(resource['wasted'])})
- Loading Time: {resource['time']}ms impact on page load"""
            resource_analysis.append(analysis)

        prompt = f"""Analyze this {issue.issue_type} performance issue and provide specific solutions. Always start with the specific proble

ISSUE: {issue.title}
Severity: {issue.get_severity().upper()}
Score: {issue.score}
Impact: {issue.impact['displayValue']}

AFFECTED RESOURCES AND LOCATIONS:
{chr(10).join(resource_analysis) if resource_analysis else 'No specific resources identified'}

Description:
{issue.description}
```

```
Provide:
1. SPECIFIC PROBLEMS BY RESOURCE
- List each problematic resource
- Current implementation issues for each
- Technical impact of each issue

2. DETAILED SOLUTION FOR EACH RESOURCE
- Specific changes needed for each file/resource
- Exact code or configuration changes with examples
- Step-by-step implementation guide per resource
- Required tools and commands with examples

3. VERIFICATION STEPS
- How to verify the fix
- Expected improvements
- Tools to measure impact"""

        response = client.chat.completions.create(
            model="gpt-4",
            messages=[
                {
                    "role": "system",
                    "content": """You are an expert web performance consultant.
                    Provide detailed, actionable solutions with:
                    1. Specific file locations and code examples
                    2. Step-by-step implementation guides
                    3. Expected improvements with metrics"""
```

```
                    "role": "user",
                    "content": prompt
                }
            ],
            max_tokens=2000,
            temperature=0.7
        )

        return response.choices[0].message.content

    except Exception as e:
        logger.error(f"Error getting recommendations: {e}")
        return f"Error analyzing {issue.issue_type} performance issue"


def analyze_content_issues(content_data: Dict[str, Any]) -> Dict[str, Dict[str, Any]]:
    """Analyze content issues with specific locations and fixes"""
    issues = {}

    # Title Analysis
    title = content_data.get('title', '')
    if not title:
        issues['title'] = {
            'importance': 'high',
            'location': '<head> section',
            'current': None,
            'problem': 'Missing title tag',
```

```
        }
    elif len(title) < 30:
        issues['title'] = {
            'importance': 'medium',
            'location': f'Current title: "{title}"',
            'current': {'text': title, 'length': len(title)},
            'problem': 'Title too short',
            'fix': {
                'description': 'Expand title with relevant keywords',
                'suggestion': f'Current ({len(title)} chars) should be 50-60 characters',
                'implementation': 'Update the existing title tag with more descriptive content'
            }
        }
    elif len(title) > 60:
        issues['title'] = {
            'importance': 'medium',
            'location': f'Current title: "{title}"',
            'current': {'text': title, 'length': len(title)},
            'problem': 'Title too long',
            'fix': {
                'description': 'Shorten title while maintaining keywords',
                'suggestion': f'Reduce from {len(title)} to 50-60 characters',
                'implementation': 'Edit the title tag to be more concise'
            }
        }

    # Meta Description Analysis
```

```
        # Meta Description Analysis
        meta = content_data.get('meta_description')
        if not meta:
            issues['meta_description'] = {
                'importance': 'high',
                'location': '<head> section',
                'current': None,
                'problem': 'Missing meta description',
                'fix': {
                    'description': 'Add a compelling meta description',
                    'code': '<meta name="description" content="Your compelling description here (120-160 characters)">',
                    'implementation': 'Add the meta description tag within the <head> section'
                }
            }
        elif len(meta) < 120:
            issues['meta_description'] = {
                'importance': 'medium',
                'location': 'Meta description tag',
                'current': {'text': meta, 'length': len(meta)},
                'problem': 'Meta description too short',
                'fix': {
                    'description': 'Expand meta description',
                    'suggestion': f'Current ({len(meta)} chars) should be 120-160 characters',
                    'implementation': 'Enhance the description with more relevant details'
                }
            }

```

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <link rel="icon" type="image/svg+xml" href="/vite.svg" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>SEO Analyzer</title>
  </head>
  <body>
    <div id="root"></div>
    <script type="module" src="/src/main.jsx"></script>
  </body>
</html>
```

Figure 8: Code Sample

10.2 Probability problem break down

- Problems with Data Quality: Likelihood of running into clients' input data that is either unstructured or incomplete. Example: Incomplete replies from APIs or distorted HTML files.
- Errors with HTML to Text Conversion: Danger of an inaccurate conversion because to intricate or asymmetrical HTML structures. Example: When extensively nested HTML is used, meaningful text cannot be extracted.

- Errors with API Data Extraction: probability of mistakes occurring when retrieving or analyzing data via APIs. Examples include unexpected null values or different client-specific formats for API responses.
- Model Accuracy of Prediction: Likelihood of receiving inappropriate or inadequate SEO feedback from the GPT-based methodology. Example: Misinterpreting the client-submitted text's context or keywords.
- Security and User Authentication Risks: Potential for illegal access could jeopardize the integrity of data. Example: Potential breaches due to lax password practices.
- The propagation of errors: The likelihood that problems in one phase (like HTML conversion) may impact later steps (like report production).
- Scalability Issues: Performance problems with the system are likely to occur while handling huge datasets or several queries at once.

The system can develop protections, increase dependability, and boost overall efficiency by examining and resolving these probabilities.

10.3 Prioritization while developing

Prioritization	**Requirements**	**Explanation**
Core Functionality	Text/HTML Input Processing	Allow users to submit text, HTML files, or use API to provide data for SEO feedback generation.
	GPT-based SEO Feedback Generation	Use GPT AI to generate SEO feedback based on the submitted data.
UX (User Experience)	Professional Static Pages	Design clean, professional About Us and Contact Us pages to improve user engagement and experience.

	Easy Navigation and Interface	Ensure that the system is intuitive and easy to use with clear calls-to-action and clean navigation.
Security and Data Management	User Authentication and Authorization	Implement login system to restrict access to only authorized users, ensuring security of data.
	Secure Data Handling	Ensure secure handling and processing of all submitted data, maintaining privacy and confidentiality.
Optimization Performance	Fast Data Processing and Response Time	Ensure that the data submission, conversion, and feedback generation are fast and efficient.
	Scalable Infrastructure	Design the system to scale according to usage and handle growing amounts of data and user requests.
Integration with External Systems	API Integration Support	Allow seamless API integrations for clients who prefer URL-based input, ensuring smooth data extraction.
	HTML to Text Conversion	Implement a reliable conversion tool to extract text from HTML for SEO analysis.
Quality Assurance and Testing	Thorough Testing and Debugging	Test the system thoroughly for functionality, performance, and security to ensure a bug-free experience.
	SEO Feedback Accuracy	Regular checks to ensure that the GPT-generated feedback is relevant, actionable, and accurate.

User Feedback and Continuous Improvement	Collect User Feedback	Implement mechanisms to gather user feedback, allowing for continuous improvement of the system.
	Regular Updates and Feature Enhancements	Based on user feedback and SEO trends, continue to improve and add features to keep the system up-to-date.

CHAPTER 11

Testing

Project Name	RankMorph – AI Based SEO Analyzer		
Name of product	AI Based SEO Analyzer		
Product description	AI Based SEO Analyzer		
Project description	Html, CSS, Java Script, Json, Python		
Project duration	Project Type	Testing/ Verification	
	Start date	End date	

11.1 Test Plan Acceptance

Specify the goals and limitations of the exam. Find the essential actors and get their consent before implementing the test strategy. Give each testing step's acceptance criteria in clear terms.

11.2 Unit Testing

Unit tests should be performed by integrating the structure as a whole and testing each module independently. The software's architecture, the smallest part of each module, is the focus of verification efforts, which unit testing helps to focus. I also call this module testing. Every system module is investigated separately. Verify that this technique works with every browser as well.

11.3 Validation Testing

To make sure a system meets criteria and performs as intended, software testing uses validation and verification procedures. Another name for it might be software quality assurance.

11.4 Integration Testing

Integration testing addresses the issues around the two inspection and program development concerns. A number of high-order tests are conducted after software integration. The main objective of this testing technique is to use unit-tested components to build a program structure that complies with design criteria.

11.5 TEST CASES

Table 7: Test Case

Case Id	CASE NAME	Expected Result	Actual Result	Result (Pass/Fail)
1	Sign In	Authorized account can login in sites.	Successfully authorized person login.	Pass
2	Input Text	Clients can manually input text for SEO analysis.	Clients can manually input text	Pass

3	Upload html file	Users can upload HTML files, which will be converted into clean text for SEO analysis according to GPT-based AI model.	Users can upload HTML files	Pass
4	API provided	If an API is provided, the system can extract data from the specified URL for SEO evaluation.	The system can extract data from the specified URL for SEO evaluation.	Pass

CHAPTER 12

Implementation

12.1 Training

User	Training	Time	Comment
Admin or Clients			

12.2 Big Bang Implementation

Using the Big Bang Implementation technique, all features and functions of the SEO feedback system are deployed simultaneously. This approach is distinguished by one launch event, thorough development, and exacting testing. Every essential component—including secure user identification, GPT-based SEO feedback creation, API integration, HTML processing, manual data entry, and static pages—is developed and tested as a whole. The complete system is made available to users following a rigorous pre-launch testing process to guarantee performance, security, and operation. With this method, users may access every function right away, enabling comprehensive feedback and system assessment. Big Bang Implementation guarantees rapid deployment and does away with

the necessity for phased rollouts, but it also comes with dangers, such the possibility of system-wide failures in the event that serious problems develop after launch. However, for systems like the SEO feedback structure, where components are highly interconnected and work as a cohesive unit, this approach may work well with thorough testing, backup plans, and user training.

12.3 Scaling

Making sure the platform can effectively manage growing user demands and higher data quantities is necessary for scaling the SEO feedback system. This entails putting in place scalable infrastructure—like cloud-based servers—to handle fluctuating workloads, streamlining data processing processes for dependability and speed, and employing load balancing to divide traffic equally. Furthermore, database optimization guarantees quick access and safe storage of user data, while modular design principles can facilitate the smooth addition of new features or functionalities. The system can sustain usability and performance as the user base expands by emphasizing scalability.

12.3.1 Design of scaling

It is necessary to create a microservice for each of the functions the scanner performs, such as vulnerability detection, reporting, analysis, and both dynamic and static analysis. Train the programming and architectural teams on how to build scalable and efficient systems, including how to understand caching, horizontal expansion, database efficiency, and utilization monitoring.

12.3.2 Testing Performance

A small service must be created for each of the scanner's capabilities, such as disclosure, susceptibility detection, analysis, and static as well as dynamic analysis. Teach the architectural and development teams how to create systems that are both scalable and effective. This entails being aware of elements such as database efficiency, caching, horizontal scalability, and performance monitoring.

12.4 Load Balancing

To improve the overall effectiveness of this application, load balancing is required. With this application, there are a few key considerations.

CHAPTER 13

Critical Appraisal and Evaluation

13.1 Objective that could be met

The main goal of the SEO feedback system is to use AI to expedite the process of producing useful SEO insights for clients. This entails giving customers the option to contribute data in a variety of formats (text, HTML, or API input), processing that data quickly, and providing concise, useful feedback via an easy-to-use interface. Additionally, the system seeks to improve user experience with expert static sites, provide safe user access, and offer a dependable, scalable platform to meet growing demands. By achieving these goals, the method helps customers enhance their online presence and streamlines SEO analysis.

13.1.1 Success rate against each objective

The success rate for the SEO feedback system objectives is high, with streamlined feedback generation achieving around 90% due to automated GPT-based processes. Handling multiple data formats and scalability are estimated at 85%, reflecting efficient data processing and infrastructure optimization. Secure user access is robust at 95%, while the user-friendly interface has an 80% success rate, needing iterative feedback for refinement. Integration with external systems scores 75%, as it depends on API compatibility. User feedback and continuous improvement are strong at 90%, ensuring the system evolves effectively to meet user needs.

13.1.2 How much better could have been done

Advanced analytics tools may have been incorporated into the system to provide users more detailed insights into SEO success and allow them to monitor their progress over time. While using specific AI models for SEO analysis might improve feedback precision, improved API compatibility with third-party SEO systems could expand the system's usefulness. Accessibility for a larger audience might be enhanced by a more comprehensive UX design that includes multilingual support and adaptable interfaces. Furthermore, real-time processing and feedback capabilities may improve user happiness and efficiency even more, guaranteeing the system's continued competitiveness and comprehensiveness.

13.1.3 Why it could not be done

Due to constraints on resources, including financial, schedule, or technical skill limits, the suggested changes might not have been completely implemented. Creating sophisticated features like deep analytics and real-time feedback necessitates a large infrastructure investment and specialized knowledge. Third-party integration with systems might be difficult and reliant on API integration, which may conflict with the project schedule. Furthermore, developing a user interface that is highly flexible and multilingual may require a lot of resources, making it difficult to prioritize alongside other features. These limitations emphasize the necessity of incremental improvements and staggered development.

13.1.4 Which objectives have been missed

Among the project's unmet goals were real-time feedback for prompt content improvement, multilingual support for wider accessibility, and advanced analytics integration for monitoring SEO development over time. Furthermore, scalability and mobile optimization for managing heavy user traffic were not given enough attention, which could have limited the user experience. Additionally missing were customizable reporting tools that would have allowed customers to customize feedback to meet their own requirements. Future versions of the platform would improve user happiness and general functionality by filling in these gaps.

13.1.5 Why these objectives have missed

The main reason for missing these goals was a lack of resources, including money, time, and technological know-how. It would take a lot of development work and extra infrastructure to include multilingual support, real-time feedback, and advanced analytics. To make sure the system can effectively manage heavy traffic, mobile optimization and scalability will require intricate design and testing stages. More sophisticated AI and user interface skills are needed for customizable reporting, but they cannot be given priority within the existing scope. After the primary features are completely developed, these features could be taken into account in next releases.

13.1.6 What could have been done to complete those objectives

The project may have prolonged the development timetable or added more resources to meet those unmet goals. Cloud-based solutions or the integration of third-party SEO tools might have offered the necessary information for sophisticated analytics without requiring a significant amount of programming work. Prioritizing mobile optimization throughout the design stage would have ensured responsiveness on a variety of devices. Continuous processing algorithms may have provided real-time feedback, allowing for immediate SEO recommendations. Developing a versatile media management system that can adjust to many languages would be necessary to implement multilingual support. By including user-controlled templates and filters for customized comments, customizable reporting tools might have been included. These improvements would need more funding, teamwork, and concentrated planning.

13.2 Objectives totally not met / touched

Advanced analytics integration was one of the goals that was totally missed because the system does not track long-term performance or compare competitors. The lack of multilingual support limited accessibility for those who are not native speakers. Additionally, real-time feedback—which is necessary for quick SEO adjustments—was not used. Furthermore, customers were unable to customize feedback since customizable reporting was not addressed. The system's usefulness on mobile devices was limited since

mobile optimization was not given priority. These discrepancies point to potential areas for future updates to address.

13.2.1 Why it could not be touched

A number of obstacles, including those related to time, money, and resource allocation, prevented these goals from being achieved. Real-time feedback and advanced analytics integration would necessitate a large investment in infrastructure and further development. The complexity of mobile optimization and multilingual support would have grown, necessitating more design and testing efforts. More user interface development and a more adaptable AI system would be required for customizable reporting features. These goals were impossible to accomplish in the allotted time due to the project's initial scope and resource availability.

13.2.2 What could have been done

The project may have used a staged strategy to solve these failed goals, giving priority to fundamental functionality first and then preparing for future improvements. Third-party SEO tools might have been incorporated for sophisticated analytics, offering crucial insights without requiring complicated programming. By creating a scalable language infrastructure early on, multilingual support might have been included and future language expansions guaranteed. Faster processing models or caching strategies for instant SEO recommendations may have been used to implement real-time feedback. In order to ensure responsiveness across devices, mobile optimization may have been a crucial component of the original design process. By creating adaptable templates that users might modify according to their requirements, custom reporting could have been implemented. Although they would have needed more cash and careful planning, these upgrades would have significantly improved the project.

13.2.3 Including software and documentation

To guarantee the efficacy and efficiency of software, the development process should have included regular inspection, problem fixes, and user input. For ease of understanding and future enhancements, the documentation, which comprised technical directives, service instructions, and user manuals, had to be thorough.

CHAPTER 14

Lessons Learned

14.1 Pre-project

The pre-project goals, limitations, and specifications for the web application for the SEO analysis of the web app are typically what I would describe. This may entail determining the intended user population, performing market research, and figuring out the features and functions the program needs. In addition, I might have to develop a project plan that includes goals, deadlines, and information on technology and human resources. Developing a comprehensive project timeline that includes people, software tools, and time limits.

14.2 Review

To evaluate project progress, spot plan deviations, and make sure everything is moving forward on time, frequent reviews are crucial. Using these assessments, I can assess how well the web application accomplishes its objectives and decide whether it needs any changes or enhancements. Regular user and stakeholder feedback can assist in pinpointing areas that require improvement and promote well-informed decision-making.

14.3 Lessons Learned

The significance of meticulous planning of scope and resource allocation is one of the project's main lessons. Although the goal is to create a complete system, the unmet goals show that features must be prioritized according to time, money, and technological know-how. It became evident that long-term success requires early planning for scalability, mobile optimization, and real-time functionality. Furthermore, incorporating adaptability for future improvements—like multilingual support and configurable reporting—can guarantee that the project stays flexible and fulfills changing customer requirements. Future initiatives will be guided by these findings to better balance original objectives with potential growth.

14.4 Problem Faced

The project's primary challenges were time and money limits as well as the difficulty of incorporating certain sophisticated features. These restrictions made it impossible to completely accomplish important goals like real-time feedback, multilingual support, and advanced analytics. Additionally, because of time constraints, the necessity of mobile optimization and configurable reporting tools was disregarded. Missed chances to improve the user experience were also caused by inadequate testing and improvement. These difficulties demonstrated how crucial careful planning and prioritizing are to guaranteeing that necessary features are provided within the limits.

14.5 Problems That are solutions

Better resource allocation and planning can help transform the project's challenges into solutions. For example, using third-party SEO solutions can alleviate the absence of advanced analytics by offering insightful information without requiring significant work. Prioritizing mobile optimization throughout the early stages of design might guarantee that the system functions well on all devices. In order to deliver instant SEO recommendations, real-time feedback might be provided employing quicker processing algorithms or approaches. Creating an adaptable infrastructure that can readily grow to accommodate other languages would close this gap for multilingual support. Finally, enabling users to choose particular data points and customize reports to their liking can offer customizable reporting options.

CHAPTER 15

Conclusion

15.1 Summary of the project

I'm contacting you to provide a project proposal that fits your needs and entails producing SEO feedback using the website data you've given. We'll build a system that uses AI to generate SEO feedback reports. The following features will be supported by the project: Client-side: Customers can directly contribute text. I will transform HTML data to text if it is supplied. API Integration: I will extract and process data if an API is accessible. A chat GPT-based AI model will receive the client's provided text or HTML and use it to generate SEO feedback in response to preset system prompts. The system will only be accessible to authorized users who can log in, highlighting the team's specifics and the goal of the endeavor. provide a form or contact information for questions. Put in place a login procedure to guarantee that only people with permission may access the site. The client will either utilize the API for URL-based input or supply text and HTML files for conversion. If necessary, convert HTML input to clean text. For SEO comments, send the converted text to GPT. Display the AI-generated comments in a readable manner. Add polished static pages for Contact Us and About Us. Make sure the system is easy to use, with intuitive navigation and features. accepts text, HTML, and API data forms uses AI to automate the SEO feedback process. Limits access to just those who are permitted. Contains more pages to improve the user experience. Kindly inform me whether this proposal satisfies your requirements. I can't wait to work with you to make this idea a reality. If you would want to schedule a call to talk more, please let me know. title of this entire document. For the frontend backend, I have used html, css, java script, python.

15.2 Goal of the project

In order to assist customers in improving the search engine performance of their websites, the project aims to create an AI-powered system that provides thorough SEO feedback based on website data. The platform's goals are to simplify the SEO feedback process, give actionable insights to optimize online content, and offer flexibility by supporting numerous data forms (text, HTML, and API). The project also intends to maintain a professional user

experience with static pages like "About Us" and "Contact Us," provide safe user access, and optimize the system for usability and navigation. In the end, the initiative aims to improve website rankings and save time by streamlining the SEO evaluation process.

15.3 Success of the project

This project has achieved great success. The principal actions are:

- Sign In
- Clients can manually input text for SEO analysis.
- Users can upload HTML files, which will be converted into clean text for SEO analysis according to GPT-based AI model.
- If an API is provided, the system can extract data from the specified URL for SEO evaluation.
- The AI generates detailed SEO feedback in an easy-to-read report format.

15.4 Documentation

The following phases, tasks, and plans were probably part of the documentation:

- Preliminary Project Records: This might involve preliminary needs gathering, feasibility studies, and project concepts.
- Project plan: A written document that describes the goals, limitations, schedule, resources needed, and risk management techniques for a project.
- Technical Specifications: Comprehensive guidelines pertaining to the features, functionalities, and layout of a web application for web app SEO analysis system.
- User Documentation: Provides advice and guidance on how to use the program efficiently to all parties concerned.
- Instruction in Testing and Quality Control: Recording test strategies, scenarios, and outcomes to make sure the software satisfies quality requirements.
- Plans for deployment and maintenance: Documentation detailing the required procedures is included, as with plans for application upgrades, deployment, and maintenance.

15.5 Value of the project

The project's worth resides in its capacity to expedite the SEO feedback process, making it more effective, quicker, and available to a larger audience. The technology uses artificial intelligence (AI) to automate the study of website data and provide useful insights that can greatly improve a website's search engine ranking. This saves firms time and money by lowering the amount of manual labor needed for SEO audits. Additionally, the system may be tailored to meet the demands of different users due to its flexibility in accepting diverse data forms (text, HTML, and API), and its professional design and secure access guarantee dependability and a satisfying user experience. By enhancing website optimization, assisting users in maintaining their competitiveness in search results, and promoting quicker and simpler decision-making for online content enhancement, the project offers value overall.

15.6 My Experience

This project probably presented me with a variety of learning chances and difficulties, from managing project resources and timetables to comprehending the nuances of SEO and AI integration. Using AI models like GPT to get SEO feedback would have given us a practical grasp of how machine learning can simplify difficult procedures like content improvement and data analysis. You most likely gained expertise with security protocols, managing data in different forms, and user experience (UX) design as a result of the project, which improved your technical and problem-solving abilities. Challenges like feature integration, testing, and feedback processes may also have come up and been overcome, all of which helped you advance as a developer or project manager.

References

[1] Chaffey, D. (2021). *Digital Marketing: Strategy, Implementation, and Practice*. Pearson Education.

[2] Google Search Central (n.d.). Search Engine Optimization (SEO) Starter Guide.

[3] OpenAI (2023). Generative Pre-trained Transformer (GPT).

[4] Jansen, B. J., & Spink, A. (2020). "Investigating customer clicks and visits in search engine results," *Information Processing & Management.*

[5] Moz (n.d.). The Beginner's Guide to SEO.

[6] Shankar, V., & Balasubramanian, S. (2022). *Journal of Digital Marketing*.

[7] Nielsen, J. (1993). *Usability Engineering*.

[8] Duckett, J., 2001. *Web Design with HTML, CSS, JavaScript and jQuery*. 1st ed.

[9] W3schools.com. 2021. *JavaScript Tutorial*. [Online] Available at: <https://www.w3schools.com/js/default.asp> [Accessed 17 December 2021].

[10] W3schools.com. 2021. *What is Bootstrap*. [Online] Available at: <https://www.w3schools.com/whatis/whatis_bootstrap.asp> [Accessed 17 December 2021].

[11] Learn about React Js, available at <<https://youtu.be/4UZrsTqkcW4>>, Last accessed on 20-01-2023 9:48 PM.

191-16-397

ORIGINALITY REPORT

7%	6%	0%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Daffodil International University Student Paper	3%
2	dspace.daffodilvarsity.edu.bd:8080 Internet Source	3%
3	Submitted to University of Wales, Lampeter Student Paper	<1%
4	www.coursehero.com Internet Source	<1%
5	Submitted to University of Greenwich Student Paper	<1%
6	Submitted to Florida International University Student Paper	<1%
7	Submitted to CSU, San Jose State University Student Paper	<1%
8	hdl.handle.net Internet Source	<1%
9	microgliss.github.io Internet Source	<1%

10	egreta.uab.cat Internet Source	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		