Faculty of Computing and
Information Technology
King Abdulaziz University
Ministry of Higher Education
Kingdom of Saudi Arabia

كلية الحاسبات وتقنية المعلومات
جامعة الملك عبد العزيز
وزارة التعليم العالي
المملكة العربية السعودية

CPIT 499 Final Report

AI-powered Arabic Marketing Web-based Application for Businesses and Individuals

MOSAWEEQ

مُسوِّق

By

Abeer Aljohani	**2005989**
Rana Alshehri	**2005625**
Jomana Sayadi	**2005725**
Lama Albelewi	**2005630**

Supervised By
Dr. Hind Almisbahi

Department of Information Technology
King Abdulaziz University
Jeddah – Saudi Arabia
[Winter 2024]

ABEER ALJOHANI

RANA ALSHEHRI

JOMANA SAYADI

LAMA ALBELEWI

THIS REPORT IS SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE BACHELOR DEGREE IN INFORMATION TECHNOLOGY

DEPARTMENT OF INFORMATION TECHNOLOGY

FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY

KING ABDULAZIZ UNIVERSITY

May 2024

DECLARATION by AUTHORS

"I/we certify that this work has not been accepted in substance for any degree and is not concurrently being submitted for any degree other than that of BS Information Technology being studied at King Abdulaziz University, Jeddah. I/we also declare that this work is the result of my/our own findings and investigations except where otherwise identified by references and that I/we have not plagiarized another's work".

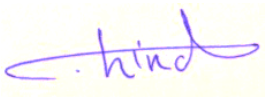



Abeer Aljohani | Lama Albelewi | Rana Alshehri | Jomana Sayadi

---

DECLARATION by SUPERVISOR

I, the undersigned hereby certify that I have read this project report and finally approve it with recommendation that this report may be submitted by the authors above to the final year project evaluation committee for final evaluation and presentation, in partial fulfillment of the requirements for the degree of BS Information Technology at the Department of Information Technology, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah.



Dr. Hind Almisbahi

Acknowledgement

We would like to extend our heartfelt gratitude to everyone who has supported and guided us throughout our senior project journey.

First and foremost, we owe our deepest appreciation to our project supervisor, Dr. Hind Almisbahi. Your continuous support, invaluable advice, and unwavering patience have been instrumental in the completion of our project, MOSAWEEQ. Your expertise has inspired us to stretch our limits and strive for excellence.

We are incredibly thankful to our families, whose unconditional love and encouragement have been our guiding light, especially during the most challenging moments. Your belief in us has been a constant source of strength and motivation, and we are forever grateful for your unwavering support.

We would also like to express our sincere thanks to our professors and the faculty of computing and information technology at King Abdulaziz University. Your guidance, insightful feedback, and academic mentorship have been essential in shaping our academic paths. Each of you has made invaluable contributions to our growth and learning, and for that, we are truly thankful.

To our friends and classmates, your friendship and support have made this journey memorable and enjoyable. The shared experiences, late-night discussions, and mutual encouragement have been invaluable. We are confident that the bonds forged through these shared experiences will continue to inspire and motivate us as we navigate the exciting paths that lie ahead.

Lastly, we are grateful to King Abdulaziz University for providing us with the necessary resources and opportunities to realize our project. From access to advanced facilities to mentorship programs, your support has enabled us to explore new ideas and bring our vision to life.

This project is a testament to the collective effort and support of everyone mentioned above. We are deeply honored to have had the opportunity to work with and learn from such amazing individuals. As we start on new endeavors, we carry with us the lessons learned and the support received, knowing that our journey ahead is enhanced by the experiences shared during this project. Thank you for your unwavering belief in us and for making this achievement possible.

Abstract

Marketing is essential for any business to succeed. It is the process of connecting products and services with their intended audience. Effective marketing can help businesses to deliver value, build brand recognition, and drive growth. In today's digital world, effective marketing is more important than ever.

The 2030 vision of Saudi Arabia is a transformative plan aimed at creating a more prosperous and inclusive society. At its core, this endeavor places a strong emphasis on fostering a thriving economy that opens doors of opportunity for all citizens. Our marketing project is deeply rooted in the principles of this program and revolves around enhancing economic capabilities through the adoption of emerging technologies. Using the power of Generative Artificial Intelligence, we offer a range of services to foster the growth of small and medium-sized enterprises (SMEs) and productive families.

Our platform provides SMEs with access to innovative digital marketing solutions, such as AI-powered content generation and branding support. This helps them to reach new customers, increase sales, and grow their businesses. We believe that our marketing project can make a significant contribution to the Saudi Vision 2030 by helping SMEs and productive families to thrive. By providing them with the tools and support they need to succeed, we can help to create a more prosperous and inclusive economy for all Saudis.

In our project, we successfully developed the MOSAWEEQ system. The system achieved its objectives in enhancing economic capabilities and supporting SMEs. However, the project encountered several limitations, such as being accessible only via desktop browsers, limited regeneration options for logos and marketing content, issues with calendar event interaction, delays in image loading, unreliable tagline generation, inaccurate image generation, and repetition in business name suggestions. Despite these limitations, MOSAWEEQ has the potential to significantly contribute to the growth and development of SMEs in line with the Saudi Vision 2030.

المُلخص

التسويق أمر أساسي لنجاح أي عمل تجاري. إنه عملية ربط المنتجات والخدمات بجمهورها المستهدف. يمكن للتسويق الفعّال مساعدة الشركات في تقديم القيمة، وبناء السمعة التجارية، وتعزيز النمو. في عالمنا الرقمي الحالي، يعتبر التسويق الفعّال أكثر أهمية من أي وقت مضى.

رؤية المملكة العربية السعودية لعام 2030 هي خطة تحويلية تهدف إلى خلق مجتمع أكثر ازدهارًا وشمولية. في جوهر هذا العمل، تولي هذه الجهود تركيزًا قويًا على تعزيز اقتصاد مزدهر يفتح أبواب الفرص لجميع المواطنين. مشروعنا التسويقي متجذر بشكل عميق في مبادئ هذا البرنامج ويدور حول تعزيز القدرات الاقتصادية من خلال اعتماد التكنولوجيات الناشئة. باستخدام قوة الذكاء الاصطناعي التوليدي، نقدم مجموعة من الخدمات لتعزيز نمو المشاريع الصغيرة والمتوسطة والأسر المنتجة.

توفر منصتنا للمشاريع الصغيرة والمتوسطة حلاً مبتكرًا في مجال التسويق الرقمي، مثل إنتاج المحتوى بقوة الذكاء الاصطناعي ودعم التسويق. هذا يساعدهم على الوصول إلى عملاء جدد وزيادة المبيعات وتنمية أعمالهم. نحن نؤمن أن مشروعنا التسويقي يمكن أن يقدم مساهمة كبيرة في رؤية المملكة العربية السعودية لعام 2030 من خلال مساعدة المشاريع الصغيرة والمتوسطة والأسر الإنتاجية على الازدهار. من خلال توفير الأدوات والدعم اللازمين لنجاحهم، يمكننا المساهمة في خلق اقتصاد أكثر ازدهارًا وشمولية لجميع السعوديين.

حقق مشروعنا نجاحًا كبيرًا في تطوير نظام مُسوِّق. وقد حقق النظام أهدافه في تعزيز القدرات الاقتصادية ودعم المنشآت الصغيرة والمتوسطة. ومع ذلك، واجه المشروع بعض القيود، مثل إمكانية الوصول إليه فقط من خلال متصفحات سطح المكتب، وخيارات محدودة لإعادة انشاء الشعارات والمحتوى التسويقي، ومشاكل في الربط مع احداث التقويم، وتأخير في تحميل الصور، وعدم الموثوقية في إنشاء الشعارات، وإنشاء صور غير دقيقة، وتكرار في اقتراحات أسماء الشركات. وعلى الرغم من هذه القيود، فإن مُسوِّق لديه القدرة على المساهمة بشكل كبير في نمو وتطور المنشآت الصغيرة والمتوسطة بما يتماشى مع رؤية السعودية 2030.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

1.1 Introduction

Marketing is essential for any business to succeed. However, creating engaging and effective marketing content can be challenging, especially for small and medium-sized enterprises (SMEs) and startups. In this chapter we explore our project MOSAWEEQ, which is a web-based marketing application that empowers businesses and individuals to create engaging Arabic marketing content that connects with their audience.

MOSAWEEQ seamlessly integrates content generation and branding support using generative artificial intelligence (AI) to simplify business growth. The platform provides a variety of tools to help businesses generate high-quality Arabic marketing content, create a strong and memorable brand identity, and schedule their content to coincide with official events and holidays.

1.2 Problem Definition

Starting a new business and attracting customers is a challenging task for many people, especially in Arabic-speaking countries, given the lack of Arabic content generation tools and the need to write effective marketing taglines, brand and design their business, and time their marketing efforts to coincide with special occasions or holidays. Businesses may also face financial challenges and time constraints when running marketing campaigns. This can be a major challenge for business owners trying to reach new customers or expand their market share.

This project aims to help businesses create marketing content using generative artificial intelligence, which can generate text and images based on user prompts. This can help businesses overcome the challenges of creating effective marketing content, such as a lack of writing or design skills, or a lack of awareness of the importance of timing their marketing efforts.

1.3 Project Scope

In line with the 2030 Vision of Saudi Arabia, the Marketing project will develop a web-based application that seeks to embrace emerging technologies, particularly generative artificial intelligence to empower Productive Families and Small and Medium-Sized Enterprises (SMEs) to start and grow their businesses by simplifying the marketing process and enabling them to create engaging and effective marketing campaigns. [1] [2]

By utilizing generative AI models, we will generate relevant Arabic text about the business domain. Additionally, we will use generative AI to produce images that facilitate the creation of attractive and unique marketing advertisements. These features can be general or tailored specifically for official events and holidays.

1.3.1 Aims and Objectives

The MOSAWEEQ project aims to develop a web-based application that will empower business owners and individuals to start and grow their businesses by simplifying the marketing process and enabling them to create engaging and effective marketing campaigns using generative artificial intelligence (AI).

The objective of MOSAWEEQ is to empower productive families, Small and Medium-Sized Enterprises (SMEs) launch successful marketing campaigns by providing tools to create engaging marketing content, build a strong brand identity, and connect with customers through relevant content timed to special events.

1.4 Proposed Solution

MOSAWEEQ is a web-based application that makes marketing simple and effective. The platform empowers businesses and individuals to create engaging Arabic marketing content that connects with their audience. We seamlessly integrate content generation and branding support using generative artificial intelligence to simplify growing the business for business owners and allow them to only focus on the quality of their products or services.

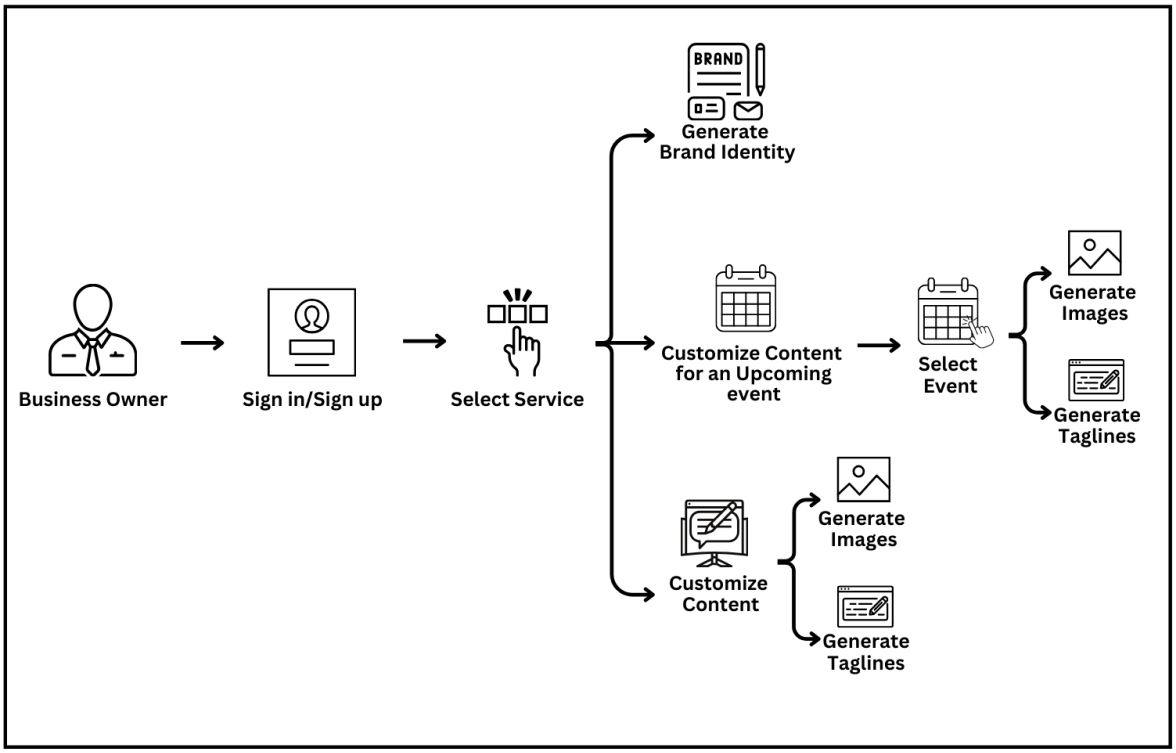


Figure 1.1: System Design

The system will include the following key features. Firstly, it will offer content generation capabilities, allowing businesses to effortlessly produce effective marketing content in Arabic. This functionality empowers businesses to create high quality marketing content, without the need for a dedicated marketing team or prior experience. Moreover, the platform will provide brand identity support, including the generation of logos and the generation of a brand name that is both strong and memorable. Additionally, the system will enable the seamless integration of marketing content with a calendar, facilitating the identification of special occasions. This integration will ensure that businesses can align their content with official events and holidays, enhancing the effectiveness of their marketing efforts. By generating Arabic marketing content, MOSAWEEQ offers several benefits to businesses, including reduced costs, increased efficiency, and improved effectiveness of marketing efforts to increase audience reach.

1.5 Project Stakeholder

The stakeholders who are involved in this project are as follows:

1. Business Owners.
   Business owners are the primary focus of this project, including Small and Medium-Sized Enterprises (SMEs), startups, and productive families. SMEs often lack marketing resources and expertise but need to connect with their target audience for growth. Startups require a quick brand identity establishment to stand out in a competitive market, and our platform aids in logo, slogan, and marketing content creation for this purpose. Productive families, with varying levels of business experience, can also benefit from our platform by enhancing their product and service marketing, thereby increasing income and productivity.

2. The developers who have administrative and development roles in the project.

1.6 Methodology

We believe that the iterative waterfall methodology will help us launch a comprehensive and effective marketing platform in a timely and efficient manner.

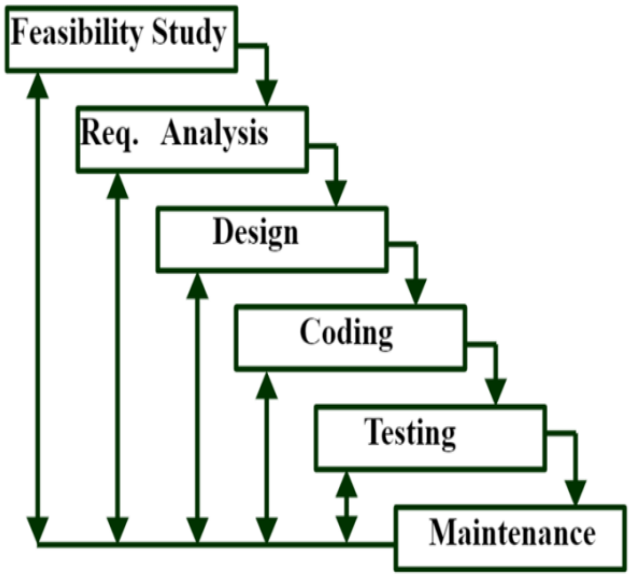


Figure 1.2: Iterative Waterfall Methodology Stages

We chose the iterative waterfall methodology, which combines the sequential nature of the waterfall model with the flexibility of iterative design, where we have the flexibility to alter and modify any phase during the development of the project and overcomes the drawback of lacking feedback from previous phases [3] This is a good fit for our project because we can work sequentially and at the end of every phase, we can review the iteration and make the necessary changes.

1.7 Gantt Chart

Phase	Start Date	End Date	Duration	Dependencies
Planning and Requirements Gathering	9/3/2023	10/5/2023	32	
Analysis and Design	10/6/2023	11/24/2023	49	Planning and Requirements Gathering
Implementation	12/26/2023	2/22/2024	58	Analysis and Design
Testing and Deployment	2/26/2024	4/11/2024	45	Implementation
Maintenance and Support	4/15/2024	5/31/2024	46	Testing and Deployment

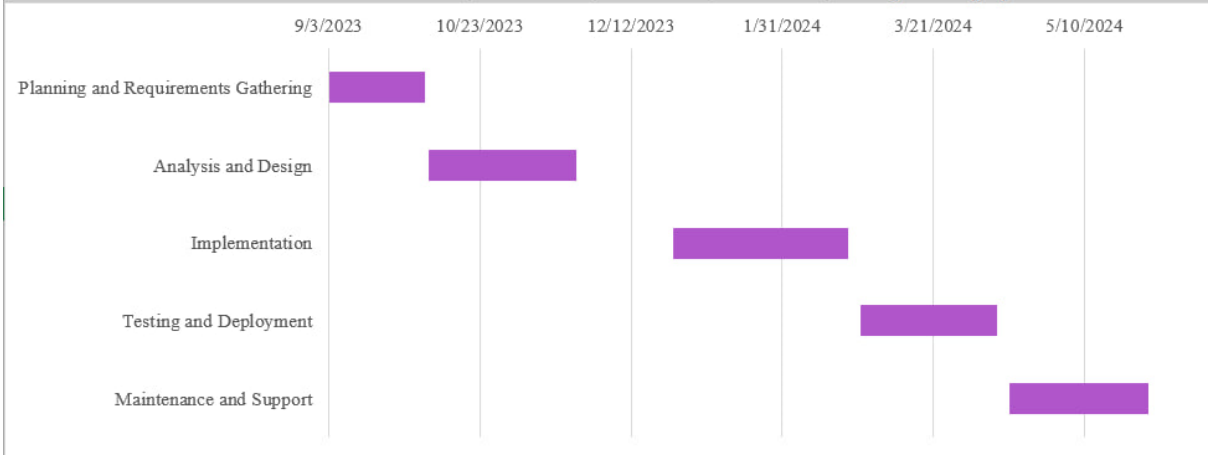


Figure 1.3: Gantt Chart

The Gantt chart serves as a comprehensive visual representation of the MOSAWEEQ project's planned timeline, spanning from September 3, 2023, to May 31, 2024. It effectively illustrates the project's breakdown into five distinct phases: Planning and Requirements Gathering, Analysis and Design, Implementation, Testing and Deployment, and Maintenance and Support. The chart also clearly depicts the dependencies between these phases, ensuring that work on subsequent phases can only commence upon completion of the preceding ones. This visual representation provides a clear overview of the project's timeline and dependencies, making it an invaluable tool for effective project management and ensuring adherence to the planned schedule.

1.8 Designing Tools

The tools that we will use while creating MOSAWEEQ are the following:

- **Project Management:** Google Drive, Notion, GitHub, and Microsoft Excel.

  Google Drive and GitHub can be used to efficiently share all project related files and documents, as well as track and maintain an online backup of many file versions. These tools can manage all project processes and files throughout the project lifecycle. Finally, Notion is used for the provided project management tools that allow us to assign tasks, create calendars, charts, and roadmap, in addition to visualizing the work on boards which helps tracking project progress and task completion [4]. Additionally, Microsoft Excel will be employed to create the Gantt chart, providing a visual representation of the project timeline and dependencies.

- **Software Analysis and Design:** Figma, Visual paradigm.

  To design the user interface (UI) and user experience (UX) of our project, we will use Figma, a design platform that allows us to create and collaborate on designs effectively. Figma provides us with a wide range of tools and features for designing UI/UX, including vector graphics editing, component-based design, and prototyping.
  Additionally, Figma's real time collaboration features make it easy for team members to work together on designs and get feedback quickly. Furthermore, in Chapter 3 [**Analysis**] and 4 [**Design**], we will use Visual Paradigm for designing the diagrams. Visual Paradigm will serve as a powerful tool enabling us to create comprehensive and illustrative diagrams essential for our project's analysis design.

- **Software Development:** HTML, CSS and JavaScript, SQLite, Flask.

  HTML, CSS, and JavaScript are the core technologies for building web pages and web applications. HTML is used to define the structure of a web page, CSS is used to style the web page, and JavaScript is used to add interactivity to the web page. SQLite is a lightweight, embedded database management system (RDBMS) that stores data directly within the web application. Flask is a Python web framework for developing lightweight and easy-to-maintain web applications. It is a popular choice for developing AI based web applications.

- **IT Technologies:** NLP, Web Development, Image Generation.

  Natural Language Processing (NLP) is a branch of artificial intelligence that deals with giving computers the ability to understand human language. It has been shown to be effective in NLP tasks using neural networks. Our project will use language models like Gemini, which is for generating text, translating languages, writing creative content, and answering.
  Web development is the creation and development of web applications that provide services to users over the internet. In this project, we will develop a web application to provide our services.
  Image generation is a field of computer science that deals with the creation of images from text descriptions. It is a challenging task, but it has many potential applications, such as creating realistic images for video games and movies or generating images to illustrate news articles and blog posts. We will use it for creating our marketing images.

1.9 Changes and Modifications

Towards the conclusion of our project, MOSAWEEQ, particularly during the design and implementation phases, we identified the necessity for modifications in each of the sequence diagrams, database structure, and ER diagram. These adjustments were deemed essential as we gained a comprehensive understanding of the entire MOSAWEEQ system.

The modification of the sequence diagrams was primarily driven by the emergence of additional process details during the implementation phase. Specifically, in the brand identity sequence, we included more required information about the user's business before generating a list of brand names using the text generation model. For further elaboration, please refer to section 4.

Additionally, we incorporated a color table into the system to facilitate the creation of brand identities for users. Therefore, this required modifications to both the database structure and the ER diagram. For more detailed information regarding the color table, please see section 4.

1.10 Conclusion

In conclusion, we have identified the pressing challenges faced by businesses and individuals in creating effective marketing content, brand identities, and strategic timing. Our project's scope is to help business owners with a comprehensive web-based application driven by generative artificial intelligence. MOSAWEEQ's mission is to simplify marketing, empower its users, and redefine the way businesses connect with their audience, bringing new Arabic marketing innovation and success.

Chapter 2: Literature Review

2.1 Introduction

In today's rapidly changing marketing landscape, businesses must stay up to date with the latest trends and innovations. This chapter provides a comprehensive overview of the latest developments in marketing, with a particular focus on the integration of artificial intelligence (AI) to create engaging content. By examining the current state of marketing in both Arabic and English, as well as exploring a range of AI-powered tools, this chapter offers valuable insights for businesses seeking to elevate their marketing strategies and achieve success in today's digital environment.

2.2 Project Background

Marketing stands at the core of business success in today's digital world. It has a profound impact on the products and services and their ability to connect with their intended audience. Business owners, especially those new to the scene, often grapple with challenges like crafting compelling marketing content, establishing a distinctive brand identity, and strategically timing marketing campaigns to correspond with special occasions and holidays. In response to these common obstacles, we've developed a solution that simplifies the marketing process, making it clear and potent. Our solution depends on building a web application that utilizes generative AI to build creative and relevant marketing content.

2.2.1 Current marketing approach in Arabic and English

Current approaches in Arabic marketing are increasingly focusing on digital channels and cultural relevance. As the Arab world's online presence continues to grow, businesses are leveraging social media, influencer marketing, and localized content to engage with their target audiences. Personalization and data-driven strategies are gaining prominence, allowing marketers to tailor their messages and products to specific Arab consumer segments. Moreover, there is a growing emphasis on respecting cultural sensitivities and traditions, ensuring that marketing campaigns resonate with the diverse and unique Arabic-speaking populations across the Middle East and North Africa. This combination of technology, cultural understanding, and data-driven insights is shaping the future of marketing in the Arabic-speaking world.

2.2.2 Artificial intelligence generated texts and images

In marketing, AI could produce various types of content, including text and images. Generative AI, employing intelligent algorithms, generates novel and engaging content. This technology is revolutionizing how marketers establish connections with their audience. Leveraging generative AI, businesses can create personalized marketing materials at scale, providing everyone with a distinct experience. Whether it's crafting compelling narratives, designing visually appealing graphics, generative AI is playing a transformative role in enhancing the effectiveness and quality of marketing efforts.

There are many AI models that can be used to build Arabic texts and images. Among them is the model [5] which can be used to generate text. This model is trained on a huge amount of Arabic text, and it can be used in many tasks such as: text summarization, Language generation, question answering, and sentiment analysis. In our project we are interested more in the language generation task. To generate images, we will use minDALL-E [6] which generates images based on text prompts. minDALL-E model is trained on 14 million image-text pairs for non-commercial purposes and can be used freely. Additionally, we will utilize Stable Diffusion models. These models

represent a significant advancement in generative AI, particularly for text-to-image generation. Unlike prior models, Stable Diffusion operates efficiently on personal computers equipped with GPUs, reducing the computational demand typically associated with this type of AI. [7] This makes it a perfect fit for our project, as we can use its capabilities to generate high-quality images directly alongside the text generation from the Gemini model. This collaboration between text and image creation will contribute significantly to achieving our project's goals.

2.3 Similar Applications

In this section, we will present many examples of marketing applications/Articles and compare them to see how our web-based application (MOSAWEEQ) differs from the others.

2.3.1 Adcreative

Summary

Adcreative is a sophisticated marketing platform designed to optimize and simplify the ad creation process. The website offers a range of compelling features to enhance marketing campaigns and make them more effective. Overall, Adcreative offers a comprehensive set of features that streamline the ad creation and management process, empowering users to create compelling marketing content and drive successful campaigns. [8]

Relevance

Content Generation:

Both Adcreative and MOSAWEEQ share a fundamental feature of content generation. Adcreative utilizes advanced AI to generate high-quality marketing content, making it accessible for businesses without a dedicated marketing team. MOSAWEEQ also focuses on generating effective marketing content.

Design Marketing image:

Both platforms offer capabilities to design marketing posts. Allows users to customize various elements to create visually appealing advertisements. Both Adcreative and MOSAWEEQ aim to provide tools and features that assist in creating compelling marketing content for successful campaigns.

Difference:

Adcreative	MOSAWEEQ
- Connects multiple ad platforms, enabling direct ad account linking, streamlining ad creative creation, and providing performance insights. - Does not support Arabic Language and calendar integration. [9]	- Distinguishes by integrating content generation and design with a calendar feature. This integration enables users to strategically time their marketing campaigns around significant events and holidays. - MOSAWEEQ supports Arabic content generation.

Table 2.1: Adcreative & MOSAWEEQ Differences

2.3.2 Microsoft Designer

Summary

Microsoft Designer is a tool for creating all types of graphics, from logos and invitations to blog banners and social media posts.

What sets Designer apart from other design tools is that it uses DALL-E 2 by OpenAI. It's a type of artificial intelligence that allows you to create images and artwork simply by entering a text description of what you want. The feature will help you find unique graphics for your projects, and thanks to advanced programming they can be very realistic. [10]

Relevance

Both Microsoft Designer and MOSAWEEQ utilize AI for generating marketing posts and logos, and they both support the Arabic language, demonstrating their commitment to catering to a diverse user base, including those who communicate and create content in Arabic. This aligns with the global nature of design and marketing, acknowledging the importance of serving users in different linguistic and cultural contexts.

Difference:

Microsoft Designer	MOSAWEEQ
- Does not support calendar integration.	- Supports calendar integration, allowing users to link calendar events with their marketing campaigns.

Table 2.2: Microsoft Designer & MOSAWEEQ Differences

2.3.3 Brand crowd

Summary

Brandcrowd is an innovative platform that specializes in the creation of logos using advanced AI technology. It provides businesses and individuals with an efficient and effective means of branding and design. With its AI-driven capabilities, Brandcrowd allows users to generate unique logos, design eye-catching posts, and create engaging videos, catering to a wide range of branding and design needs. [11]

Relevance

The relevance between Brandcrowd and MOSAWEEQ lies in their shared focus on simplifying branding and design processes through the utilization of AI. Both platforms leverage AI capabilities to assist startups in designing a unique and suitable brand identity for their business. In addition to supporting the creation of Arabic and English logos. Brandcrowd specializes in logo creation providing a broader spectrum of design services. On the other hand, MOSAWEEQ emphasizes generating brand identity such as name and logo.

Difference:

Brandcrowd	MOSAWEEQ
- Primarily focuses on branding services. It provides tools for creating logos and brand identities. - Does not support calendar integration and the generation of Arabic content for social media posts.	- Focuses on marketing content generation and simplifying the marketing process by integrating with the calendar.

Table 2.3: Brandcrowd & MOSAWEEQ Differences

2.3.4 SocialBu

Summary

SocialBu is an innovative social media management tool designed to streamline and enhance content creation and distribution across various social media platforms. With a focus on ease and efficiency, SocialBu offers an AI-powered Post Generator that allows users to effortlessly create versatile and engaging content for their social media channels. The AI-powered features include an AI Post Generator, AI Caption Generator, a Prompt Generator for Text2Img, AI Blog Image Generator, and AI Quote Image Generator. By leveraging the power of AI, SocialBu empowers users to conquer content creation worries and elevate their social media presence. [12]

Relevance

The relevance between SocialBu and MOSAWEEQ lies in their shared focus on simplifying and enhancing content creation. Both platforms utilize AI capabilities to aid users in generating content efficiently, in addition to supporting the Arabic language. SocialBu's AI-powered Post Generator, Caption Generator, and various other generators align with MOSAWEEQ is emphasis on content generation, including marketing lines and slogans. The AI-driven content generation features offered by both platforms contribute to a more efficient and impactful approach to managing and amplifying an organization's online presence.

Difference:

SocialBu	MOSAWEEQ
- Primarily focused on social media management, scheduling, and engagement. - Streamlines scheduling and publishing of posts, enables responding to messages and comments without platform switching, automates social media tasks, monitors mention, and provides in-depth insights into social media performance. - Does not support calendar integration.	- Provides calendar integration with content generation and design to generate relevant content that includes the special events and holidays.

Table 2.4: SocialBu & MOSAWEEQ Differences

2.3.5 Predis.ai

Summary

Predis.ai is an AI-powered marketing tool focused on assisting businesses in creating compelling video and image content for social media platforms. The tool provides AI-powered content analysis, allowing businesses to generate engaging videos, images, and social media posts. Predis.ai aims to streamline content creation through the integration of artificial intelligence, providing businesses with the means to create captivating visuals and effectively analyze their content. [13]

Relevance

The relevance between Predis.ai and MOSAWEEQ lies in their common objective of simplifying content creation using artificial intelligence. Both platforms leverage AI capabilities to aid businesses in image generation.

Difference:

Predis.ai	MOSAWEEQ
- Provides competitor analysis insights and suggests AI-powered post ideas to maintain a competitive edge. - Offers calendar management for content scheduling on social media platforms. - Does not support Arabic and calendar integration. [14]	- Provides calendar integration with content generation and design to generate relevant content that includes the special events and holidays. - MOSAWEEQ supports Arabic content generation.

Table 2.5: Predis.ai & MOSAWEEQ Differences

2.3.6 Applications of Generative AI in Business

Summery

This article explores the applications of generative AI in business. It discusses how generative AI can be used for content creation, such as generating emails, social media posts, and product descriptions, as well as creating visual branding and advertising visuals. The article also highlights the benefits of generative AI. It mentions the use of generative AI in customer service, marketing, and search engine optimization. The article includes insights from Valossa, an AI developer, on their experimentation with generative AI and its potential applications in various fields. Overall, generative AI is seen as a powerful tool with the potential to revolutionize business operations. [15]

Relevance

Both the MOSAWEEQ and the article discuss the applications of generative AI in marketing and content creation. Both acknowledge the challenges encountered by businesses, especially small and medium-sized enterprises, in creating compelling marketing materials. Furthermore, they both underscore the objective of utilizing AI to simplify marketing processes and enable businesses to craft engaging and effective campaigns, whether through creating marketing content and to generate text and images based on user prompts. The common thread is their shared recognition of AI as a transformative tool in marketing and business growth.

Differences:

Article1	MOSAWEEQ
- The article provides a general overview of generative AI technology and its evolution, with a focus on the advancements in large language models like ChatGPT. - It discusses the use of generative AI in data analysis and customer service, among other areas. - The article discusses generative AI in a broader context, without a specific focus on Arabic content.	- MOSAWEEQ is described as a web-based marketing application designed to empower businesses and individuals in creating Arabic marketing content. - It aims to provide tools for content generation, branding support, and integration with a calendar to align marketing efforts with special occasions.

Table 2.6: Article1 & MOSAWEEQ Differences

2.3.7 The Impact of Generative AI on the Future of Visual Content Marketing

Summery

This article discusses the significance of generative artificial intelligence (AI) in shaping the future of visual content marketing. It highlights the importance of visually appealing content in today's marketing landscape and the increasing role of AI in revolutionizing the field. The integration of visual content with AI is seen as a key factor in acquiring and retaining loyal customers. The article emphasizes that the absence of AI in a company's marketing strategy could result in a smaller market share. It also explores the concept of generative media, which involves using AI algorithms to generate content based on parameters or algorithms provided by human artists. This approach not only revolutionizes the art market but also provides a more personalized experience for viewers. The article emphasizes the need for marketers to embrace emerging technologies and close the gap between traditional and technological forms of marketing. It concludes by highlighting the future of visual marketing, which includes greater personalization, real-time operations, and an increased reliance on AI and data-driven decision making. [16]

Relevant

The MOSAWEEQ project and the article on the impact of generative AI on visual content marketing share a common focus on the use of artificial intelligence in marketing. Both highlight the importance of visually appealing content and the integration of AI to enhance marketing strategies.

Relevance between MOSAWEEQ and the article:

Both emphasize the significance of visual content in marketing and the need for businesses to create engaging and high-quality content.

They recognize the potential of generative AI in automating content generation and enhancing creativity in marketing.

Both acknowledge the importance of personalization and the use of AI to tailor content to official events and holidays.

Differences:

Article2	MOSAWEEQ
- The article provides a broader overview of the impact of generative AI on the future of visual content marketing, discussing its potential in revolutionizing the art market and providing personalized experiences. - The article discusses the broader implications of generative AI in marketing.	- The MOSAWEEQ project specifically focuses on empowering businesses and individuals, particularly in Arabic-speaking countries, to create effective marketing content using generative AI.

Table 2.7: Article2 & MOSAWEEQ Difference

2.4 Comparative Study

In Table 2.6 we perform a comparative study for all the previous reviewed applications and MOSAWEEQ.

	AdCreative.ai	**Microsoft Designer**	**BrandCrowd**	**SocialBu**	**Predis.ai**	**MOSAWEEQ**	**Article1**	**Article2**
Generate Post (Images)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
Generate Text	Yes	Yes	No	Yes	No	Yes	Yes	Yes
Generate Logo	No	Yes	Yes	No	No	Yes	No	No
Connect with Calendar	No	No	No	No	No	Yes	No	No
Support Arabic Language	No	Yes	Yes	Yes	No	Yes	No	No

Table 2.0.8: Comparative Study

2.5 Conclusion

This chapter highlights the importance of cultural relevance, personalization, and data-driven strategies in Arabic marketing, along with the transformative impact of AI-powered tools like Adcreative, Microsoft Designer, Brandcrowd, SocialBu, and Predis.ai in simplifying content creation and enabling businesses to attract diverse audiences.

Chapter3: Analysis

3.1 Introduction

In a world increasingly driven by digital presence and marketing, the functionality of any business platform is fundamental. This chapter explores the functional and non-functional requirements that underscore the design and development of MOSAWEEQ, our transformative web-based application. These requirements provide a blueprint for the project's capabilities, ensuring it meets the needs and expectations of its users to simplify their marketing efforts and create compelling content.

3.2 An Interview-Driven Approach to MOSAWEEQ's Functional Requirements

We utilized a targeted interview approach to extract valuable insights directly from a key stakeholder within our team—a family member who operates an embroidery business. This interview served as an essential foundation for identifying the functional requirements essential for the development of MOSAWEEQ's web-based application.

The interview, detailed in **Appendix A**, provided invaluable qualitative data that highlighted specific pain points faced during the marketing of the embroidery business. It illuminated the necessity for a tool that could simplify marketing processes, assist in brand identity development, and streamline content creation, catering specifically to the needs expressed during the interview. These direct experiences and insights formed the defining functional requirements for MOSAWEEQ, ensuring its capabilities aligned with the identified challenges faced by business owners in the marketing.

3.3 A Survey-Driven Approach to MOSAWEEQ's Functional Requirements

To gain a comprehensive understanding of the challenges faced by business owners and individuals who are looking to start a new business, we conducted a survey and analyzed the results of the survey to identify the functional requirements for the web-based application of MOSAWEEQ. The survey, detailed in **Appendix A**, gathered insights from a diverse group of individuals involved in various stages of business ownership and entrepreneurship. The responses revealed a range of common obstacles, including difficulties in marketing and branding, content creation and the production of relevant and timely content. These insights guided the selection of functional requirements for the MOSAWEEQ platform to effectively address the identified challenges and empower businesses to grow.

3.4 Functional Requirements

Functional requirements capture the intended behavior of a system by defining the services, tasks, or functions that the end user requests explicitly. They describe what the system must do in order to meet the needs of its users. [17]

- The user shall be able to sign in with their existing account credentials.
  Justification: User authentication is essential for security and personalized user experiences. It ensures that users can access their own data and settings securely. By allowing users to sign in, the system can authenticate their identity and provide them with the correct access privileges.
  ID: F1

- The user shall be able to create a new account if they do not have an existing account.
  Justification: User sign up is essential to attract new users and grow the user base of the system. By allowing users to sign up, the system can provide them with access to all of its features and functionality.
  ID: F2

- The user shall be able to generate a brand identity for their new business.
  Justification: This feature simplifies branding and identity establishment for those who are starting new businesses.
  ID: F3

- The user shall be able to customize new content.
  Justification: This feature provides users with the essential content they need to effectively market their products and services, all within the same system.
  ID: F4

- The user shall be able to generate marketing Taglines.
  Justification: This is the core function of the web-based application, providing users with the engaging content they need to market their products and services effectively without the need to hire a marketing agency.
  ID: F5

- The user shall be able to generate marketing images for posts.
  Justification: This feature enables users to create visually engaging marketing content, which is essential for capturing the audience's attention in the digital landscape nowadays.
  ID: F6

- The user shall be able to customize new content for an upcoming event on the calendar.
  Justification: This feature enables users to see what events are coming up and plan their marketing content accordingly. By selecting from official upcoming events, the relevance and timeliness of the marketing content will be enhanced.
  ID: F7

3.5 Non-Functional Requirements

Non-functional requirements specify criteria used to assess a system's performance and determine the quality aspects of its features. [17]

- Usability and User Experience.
  The system shall have a user interface that is easy to use, navigate, and aesthetically pleasing for users of all skill levels.
  Justification: The web-based application should be designed to be user-friendly, even for users with no experience with marketing or AI. This will make it accessible to a wider range of small businesses.

- Cross-platform compatibility.
  The system shall be responsive and work on all devices.
  Justification: This requirement makes the system more accessible to a wider range of users.

- Data Privacy and Security.
  The system shall be secure and protect user data from unauthorized access.
  Justification: It is important to protect user data, especially when it comes to financial information and customer lists. The web application should implement security measures to protect user data from unauthorized access and theft.

- Interoperability.
  The system shall support integration with third-party services and APIs, allowing users to connect to various external tools and platforms.
  Justification: This requirement allows supporting integration with third-party services and APIs which provide users with the flexibility and extensibility they need to enhance their marketing efforts by allowing them to seamlessly connect with external tools and platforms.

3.6 Initial Design / Analysis

3.6.1 Use Case Diagram

A use case diagram is a Unified Modeling Language (UML) diagram that shows the interactions between users and a system. It is a graphical representation of the different ways that a system can be used, and the different types of users who can interact with it.

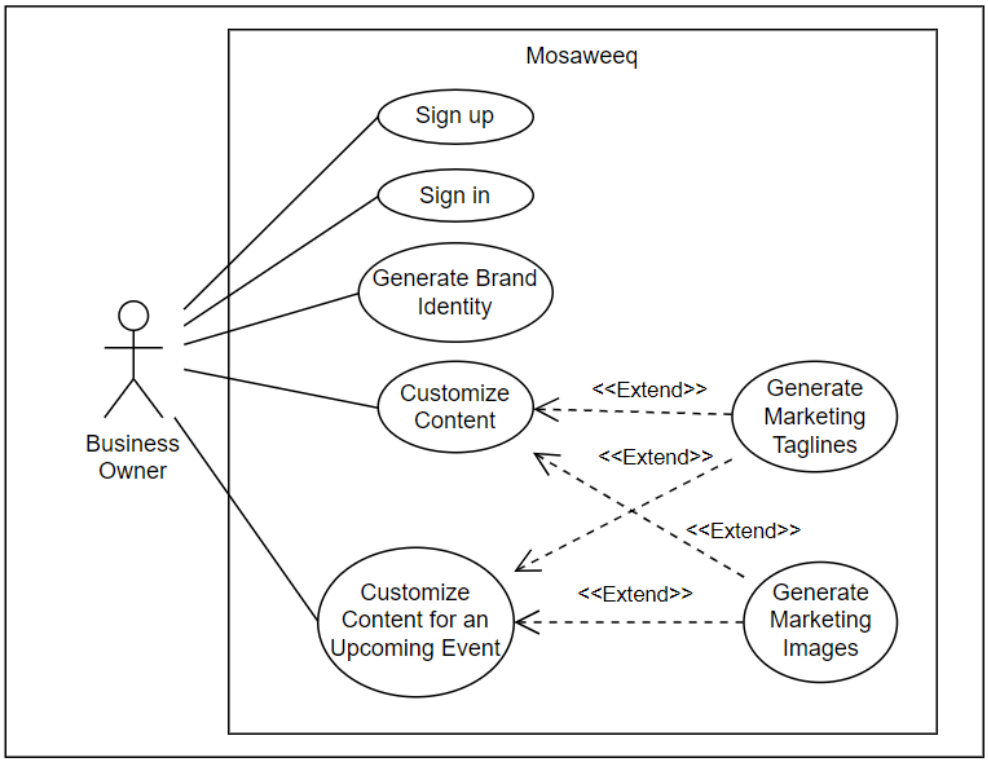


Figure 3.1: Use Case Diagram

3.6.2 Use Case Diagram Description

In Table 3.1, we will demonstrate the description of each use case.

ID	Name	Actor	Description	Preconditions	Basic Flow
U1	Sign up	Business Owner.	Business owners can create a new account to be able to use the features of MOSAWEEQ.	User does not have an existing account.	1. User accesses the sign-up interface. 2. User provides the necessary details (name, email, and password, etc.). 3. The system checks if the email address is already in use. 4. If the email address is not in use, the system creates an account for the user and sends them a confirmation email. 5. User clicks on the confirmation link in the email. 6. User is redirected to the homepage and is now logged in.

U2	Sign in	Business Owner	Registered business owners can access MOSAWEEQ web-based application from the sign-in interface.	User must have an existing account in the web-based application.	1. User accesses the sign-up interface. 2. User provide the necessary details (email, password) 3. Database stores the information in the system. 4. User is redirected to the homepage.
U3	Generate Brand Identity	Business Owner	If the user wants to start a new business this feature can generate and establish their brand identity easily.	The Business owners must have an account in MOSAWEEQ and select the generate brand identity feature to easily establish the identity for their new business.	1. User accesses the generate brand identity interface. 2. User provide the necessary details (description of the brand, category). 3. The system generates suggested brand identity. 4. Users select their preferred brand identity. 5. The system saves the brand identity and returns the user to the homepage.
U4	Customize Content	Business Owner	Business owners can request creative and relevant content for marketing.	The Business owners must have an account in MOSAWEEQ and select the customize marketing content feature to generate creative and relevant marketing content.	1. User accesses the customize content interface. 2. User enters a description of the desired content. 3. User chooses between generate marketing taglines or generate marketing images.
U5	Customize Content for an Upcoming Events	Business Owner	Business Owners can check official upcoming events and adapt the marketing content to align with them.	The Business owners must have an account in MOSAWEEQ and select from official upcoming events feature to align the marketing content with the special occasions.	1. User access official upcoming events interface. 2. The system displays a calendar for date selection. 3. User selects the event that they want to generate new content for. 4. User enters a description of the desired content. 5. User chooses between generate marketing taglines or generate marketing images.
U6	Generate Marketing Taglines	Business Owner	Business Owners can request to generate engaging taglines for marketing without external help.	The Business owners must have an account in MOSAWEEQ and select the generate marketing taglines feature to generate attractive marketing taglines.	1. User access marketing taglines interface. 2. User provide the necessary details (description of the product or service). 3. The system generates many marketing taglines. 4. User can choose between the suggested marketing taglines or regenerate if they don't prefer any of the suggestions.
U7	Generate Marketing Images	Business Owner	Business owners can request to generate visually appealing marketing images for social media posts.	The Business owners must have an account in MOSAWEEQ and select the generate marketing Images feature to generate engaging marketing images to post.	1. User access marketing images interface. 2. User provide the necessary details (description or photo of the product or service). 3. The system generates many images. 4. User can choose between the suggested marketing images or regenerate if they don't prefer any of the suggestions.

Table 3.1: Use Case Description

3.7 Mapping Requirements

Table 3.2 demonstrates the mapping matrix of the functional requirements and the use case.

	Sign in	Sign up	Generate Brand Identity	Customize Content	Generate Marketing Taglines	Generate Marketing images	Customize Content for an Upcoming Events
F1	✓						
F2		✓					
F3			✓				
F4				✓			
F5					✓		
F6						✓	
F7							✓

Table 3.2: Mapping Matrix

3.8 Conclusion

In conclusion, in this chapter we have established the fundamental requirements for the development of MOSAWEEQ, ranging from user authentication to content generation, reflecting the project's commitment to enhancing the marketing journey for business owners. Non-functional requirements emphasize usability, cross-platform compatibility, data privacy, security, and interoperability to create a secure and user-friendly environment. These requirements collectively shape MOSAWEEQ's capabilities and user experience, setting the stage for a transformative web-based application that empowers businesses to streamline their marketing efforts, in addition to guiding the design and implementation.

Chapter4: System Design

4.1 Introduction

In the fourth chapter of our MOSAWEEQ marketing project, we started our designing phase, specifically focusing on the Object-Oriented (OO) approach. This decision was driven by the project's diverse features like content generation, branding, and calendar integration. The OO approach allows us to model these features as encapsulated objects, promoting code reusability and modularity. This chapter explores different diagrams for showing the project in a better picture which helps every member understand the sequence of processes in the project clearly.

4.2 Design Approach

We decided to follow the OO approach for our project MOSAWEEQ, because it involves multiple features and interactions, such as content generation, branding, and calendar integration. Object-oriented design allows us to model these features as objects and encapsulate their behavior, making it easier to manage and maintain the system. Also, the OO approach promotes code reusability and modularity, which is essential for a project like MOSAWEEQ. By defining classes and objects, we can create reusable components, reducing development time and effort. This is particularly valuable in a web-based application where various aspects of content generation, branding, and calendar integration need to work together seamlessly. Additionally, OO design aligns well with the iterative waterfall methodology mentioned in the project's methodology. It allows for flexibility in making changes during the development process, which is essential for adapting to evolving project requirements and ensuring that the system meets the needs of business effectively.

4.2.1 Sequence Diagram

In our system, these diagrams illustrate key use cases within MOSAWEEQ: Customize Content, Customize Content for an Upcoming Event, Generate Brand Identity. Each diagram outlines step-by-step actions taken by the system and the user, from initiating a process to the outcome. They provide a clear understanding of how users navigate through the platform's features.

Customize Content Sequence Diagram:

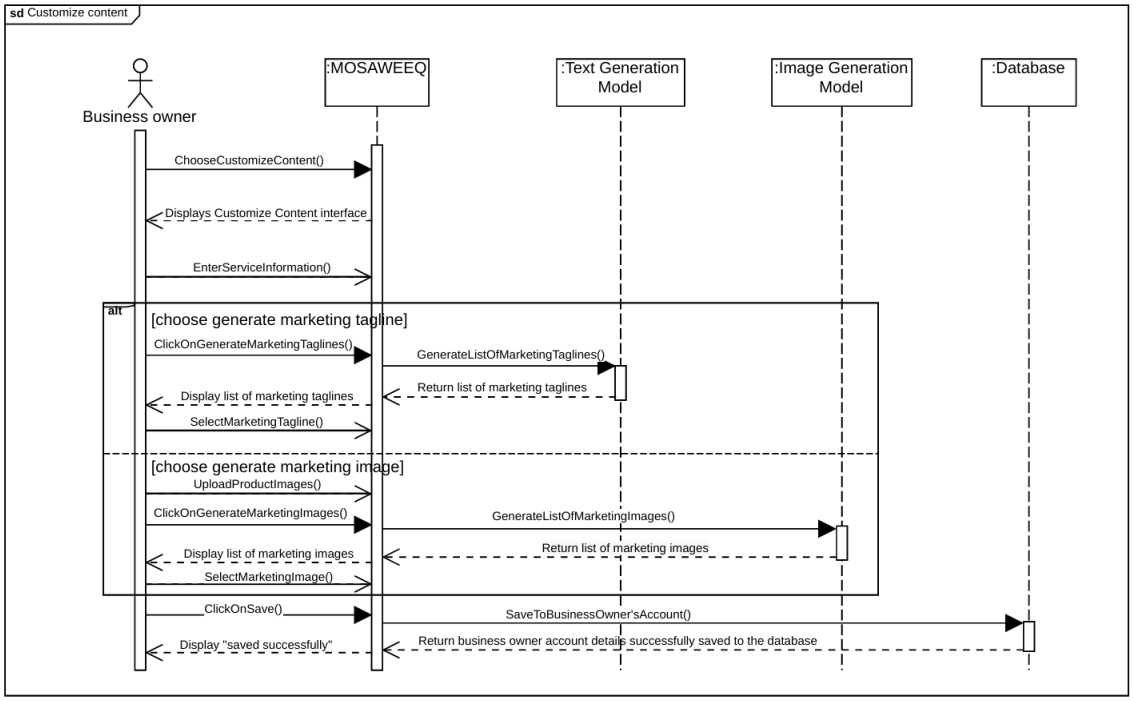


Figure 4.1: Customize Content Sequence Diagram

The sequence diagram in Figure 4.1 illustrates the 'Customize Content' use case within the MOSAWEEQ platform. It begins with the business owner choosing 'Customize Content,' which displays the customized content interface. Following this, MOSAWEEQ requests product or service description details, and then the business owner enters the product or service information. The user is presented with two choices: 'Generate Marketing Tagline' or 'Generate Marketing Image'. Opting for the marketing tagline initiates a request to a text generation model for a list of taglines, which are then presented to the business owner. The business owner selects their preferred marketing tagline. On the other hand, choosing a marketing image starts with the business owner uploading an image, which then initiates a request to an image generation model. The model generates several image options based on the description or uploaded image, and these options are presented to the business owner. Upon completing this process, the business owner can save the customized content in their account by clicking 'Save'.

Customize content for an Upcoming Event Sequence Diagram:

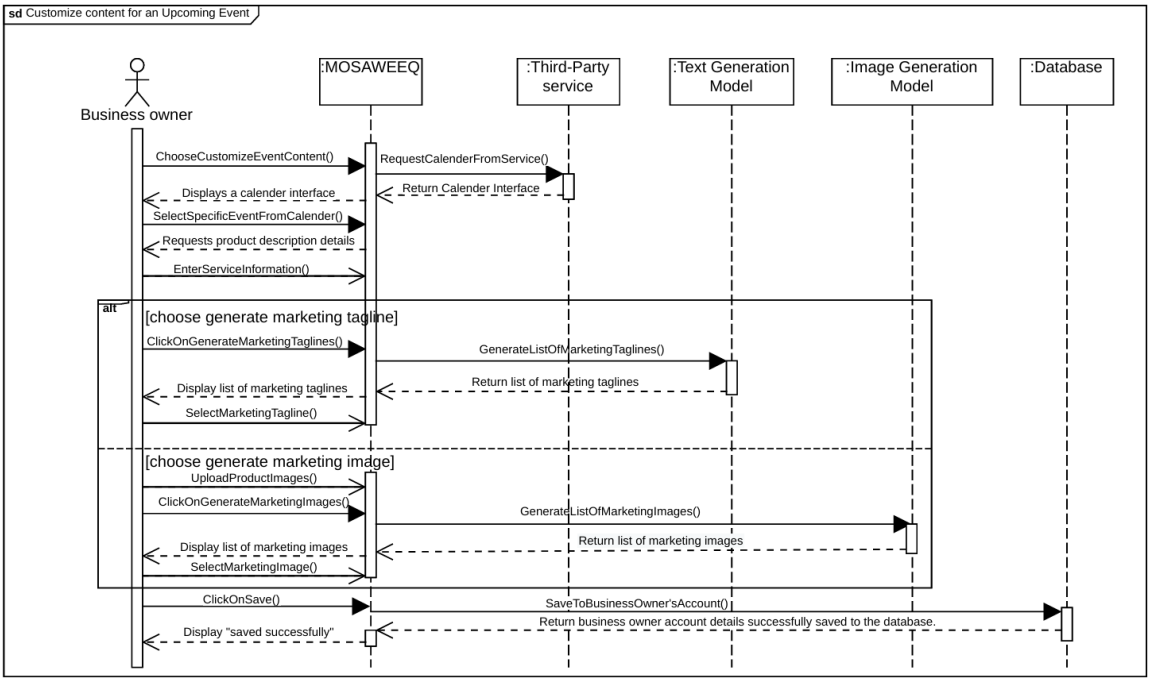


Figure 4.2: Customize content for an Upcoming Event Sequence Diagram

The sequence diagram in Figure 4.2 illustrates the 'Customize Content for an Upcoming Event' use case within the MOSAWEEQ platform. It begins with the business owner choosing 'Customize Event Content,' which triggers MOSAWEEQ to fetch and display the calendar retrieved from a third-party service. Following this, the business owner selects a specific event from the displayed calendar, prompting MOSAWEEQ to request product or service description details from the business owner. The business owner inputs the service information. At this stage, the business owner is presented with a choice between 'Generate Marketing Tagline' or 'Generate Marketing Image.' Opting for a marketing tagline prompts the display of options from a text generation model, enabling the business owner to select one. On the other hand, selecting a marketing image involves uploading an image, followed by options generated by an image generation model, from which the business owner makes a selection. Finally, the selected generated content is saved in the business owner's account by clicking 'Save.'

Generate Brand Identity Sequence Diagram:

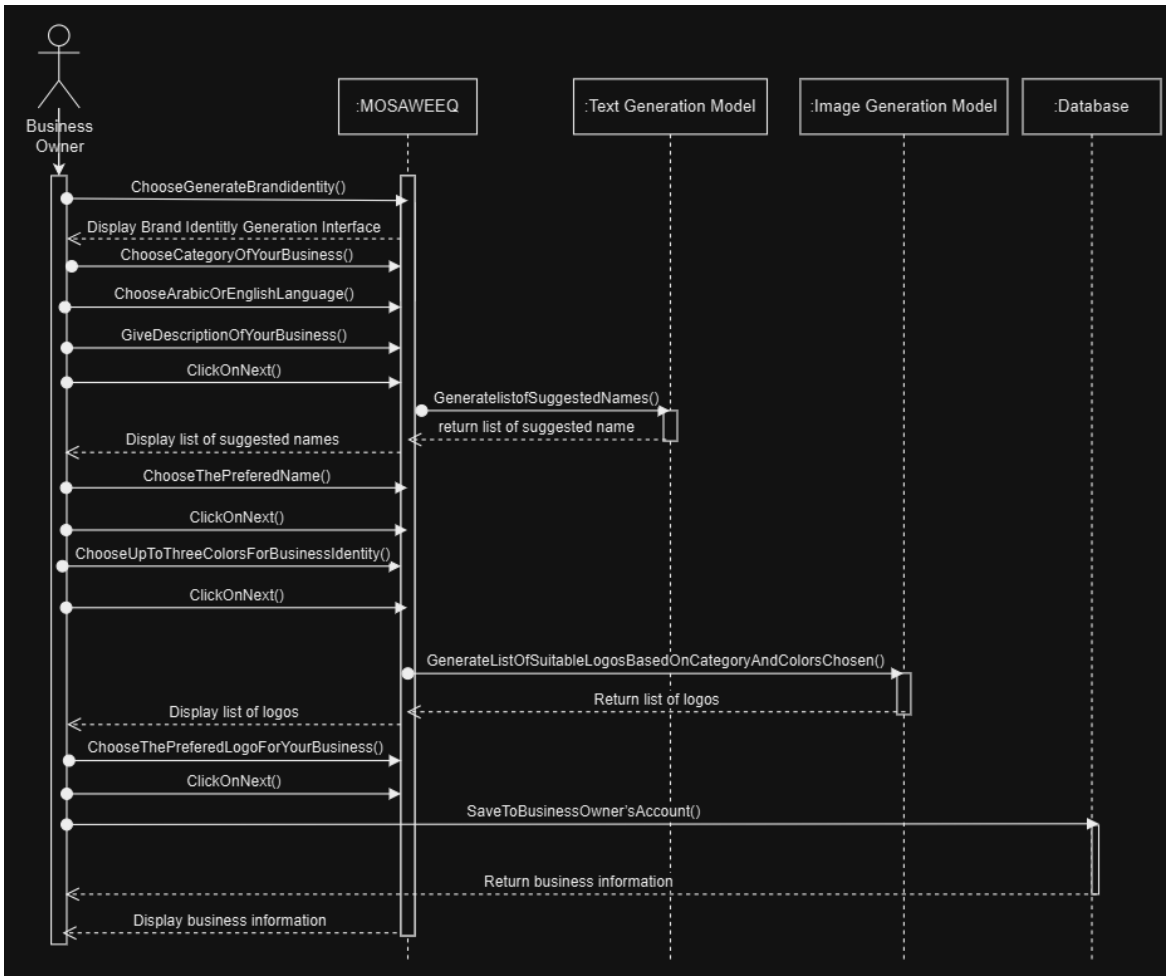


Figure 4.3: Generate Brand Identity Sequence Diagram

The sequence diagram in Figure 4.3 illustrates the "Generate Brand Identity" use case within the MOSAWEEQ platform. It begins with the business owner choosing "Generate Brand Identity", leading to the display of the service page. Subsequently, the business owner is requested to enter a service or product description of their business, choose a category for the business, and select the preferred language for the brand name (either English or Arabic). The text generation model then produces a list of suitable names, presenting them to the business owner for selection. After choosing a preferred name and clicking 'generate logo', MOSAWEEQ, in turn, requests suitable logos from Image generation model, presenting a list of options to the user. Finally, the business owner selects one of the logos, and MOSAWEEQ displays the whole brand identity, which is then saved in the business owner's account by clicking on 'save.'

4.2.2 Class Diagram

The following Figure 4.4 shows the static structure of MOSAWEEQ class diagram. A class diagram is an essential tool in object-oriented modeling that outlines the classes, attributes, and methods of a system, enabling developers to effectively communicate and implement the system's design. In the context of the MOSAWEEQ project, the class diagram is crucial for understanding the relationships between the main components.

The User class represents individuals using the platform and contains attributes that store their information and operations to deal with these attributes, while the Business class encapsulates registered businesses' information. The BrandIdentity class contains attributes that handles the visual identity of the business, and the Category class contains attributes that classifies the business to specify the generated content. The Content class represents user-generated content, it contains attributes and operations that handles the generated-content data, and the Event class represents scheduled events associated with businesses. The relationships shown in the class diagram represents that a user can have multiple businesses,and the business can have only one category and one brandIdentity, The Content and Event classes share a many-to-many association, allowing multiple Content records to be linked to various Events, and vice versa.

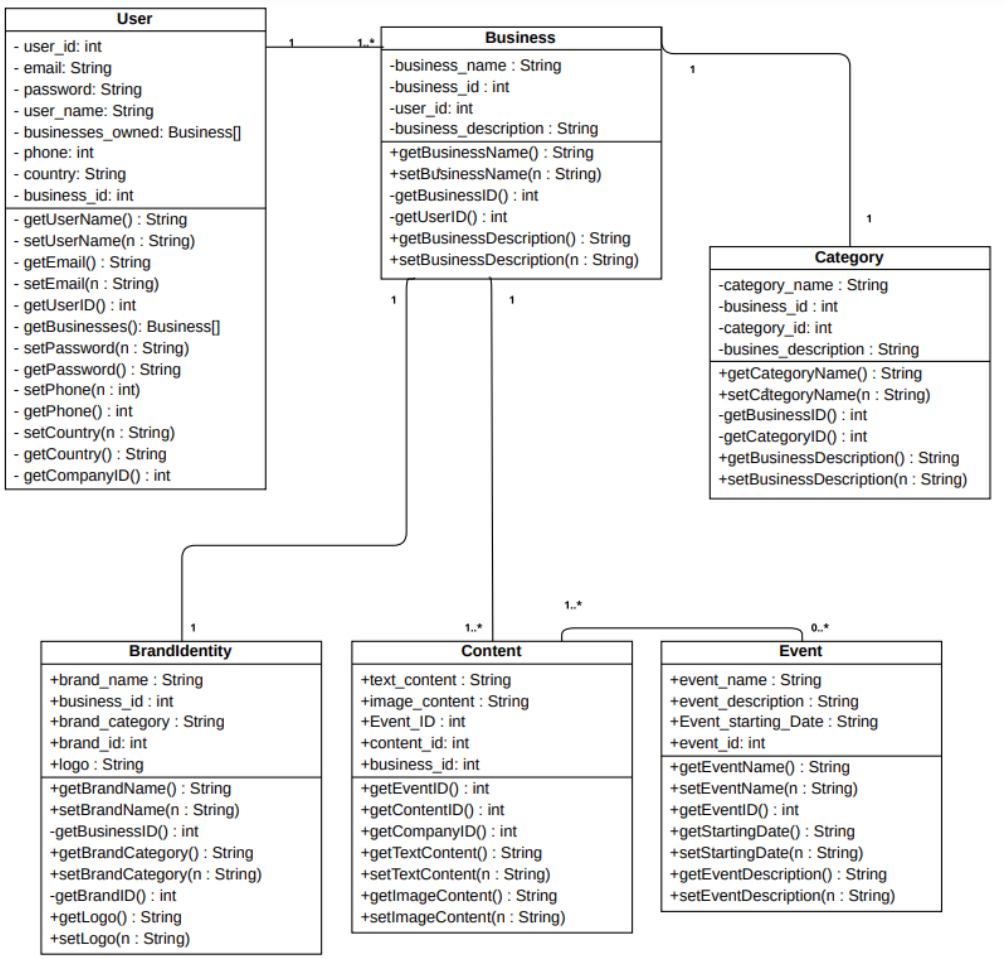


Figure 4.4: Class Diagram

4.3 Data Model Design

Designing the data model is a fundamental step in creating a database system. It helps us understand how data is structured and how it's connected. This design process relies on two primary techniques: the Entity Relationship Diagram (ERD), which shows the entities of the database and how they relate to each other [18]. The other technique is called normalization, which is a way to make sure we don't have the same data in too many places, keep things consistent, and organize everything neatly in the database [19]. Effective application of these techniques is important for a well-functioning database system, minimizing errors and providing a solid foundation for all database operations.

4.3.1 Entity Relationship Diagram (ERD)

The Entity-Relationship Diagram (ERD) of MOSAWEEQ in Figure 4.5 shows the roles of key entities in the system. The "User" entity stores user details, "Business" manages business data, "BrandIdentity" handles identity of the brand of the business, and "Category" deals with the categorization of the business to specify the generated content. The relationships in the ERD are carefully defined: User and Business have a one-to-many connection, signifying that a single user can have multiple Business records in the "Business" table; Business, BrandIdentity, and Category have one-to-one links,This signifies that each business is associated with a singular BrandIdentity and Category record. Additionally, the 'BrandIdentity' table has a one-to-many relationship with the 'Colors' table, implying that each brand identity may have multiple associated color records. The Business-Content relationship is one-to-many, allowing every business to have various generated content records. Lastly, Content and Event share a many-to-many association. This configuration signifies a flexible and dynamic association where multiple content records can be related to various events, and vice versa.

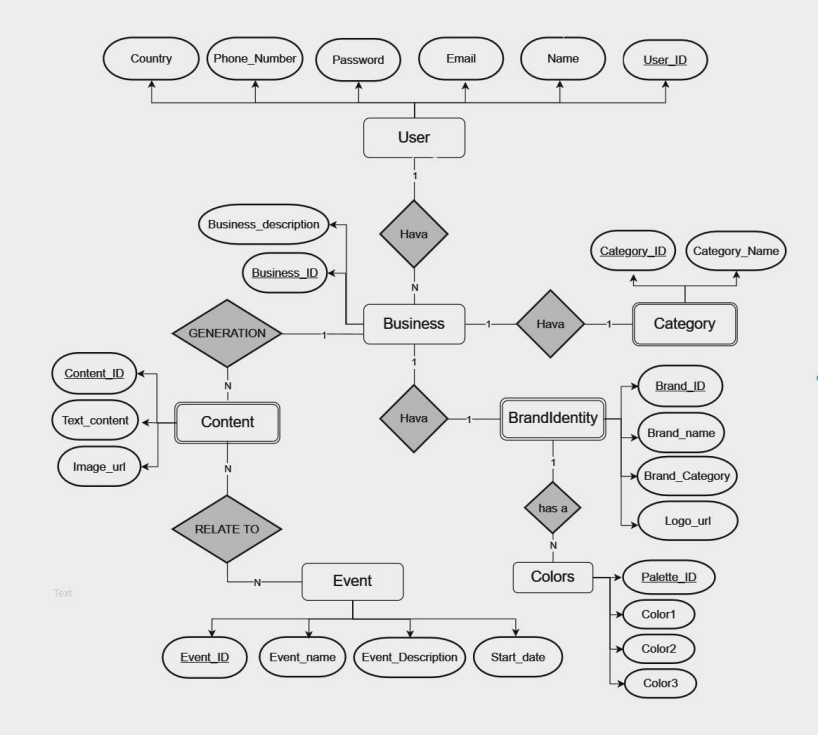


Figure 4.5: Entity-Relationship Diagram

4.3.2 Normalized Relational Schema

Our database schema comprises multiple tables that represent various components of our system, including User, Business, BrandIdentity, Colors, Category, Content, and Event. Each table serves a distinct purpose: for instance, the User table stores user-specific details, the Business table contains business-related data, and the BrandIdentity table manages brand-specific information. Relationships among these tables are established through key identifiers, connecting relevant data across different entities. Upon Comprehensive review, the schema displays a well-organized structure. Importantly, it avoids repeating groups, transitive dependencies, and partial dependencies. This careful design enhances data integrity, streamlines operations, and supports the reliability of stored information in the database.

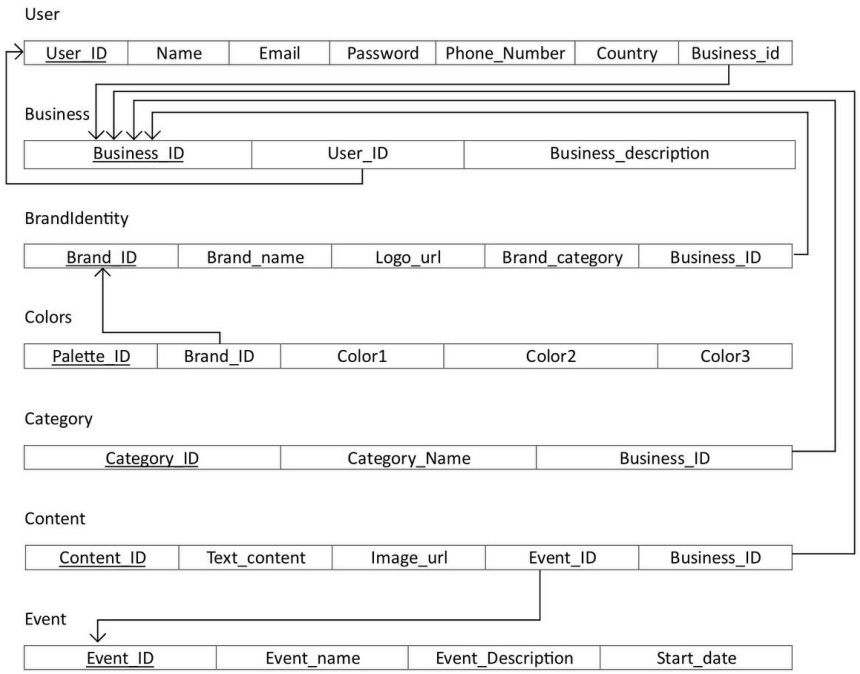


Figure 4.6: Relational Database Schema

4.4 Conclusion

In summary, the OO design significantly influences MOSAWEEQ's dynamic features. The diagrams of content and event customization, along with the Generate Brand Identity diagram, offer a comprehensive insight into the sequential flow of the system's processes. These diagrams effectively illustrate the steps involved in customizing content either for specific events or general purposes and generating a unique brand identity for businesses. Additionally, the MOSAWEEQ class diagram and data model design steps ensure a solid foundation for a resilient, adaptable, and well-organized marketing application.

Chapter 5: Implementation

5.1 Introduction

In this chapter, we will explain the implementation of Mosaweeq, and cover the construction of both the front end and back end, and detail how they are seamlessly integrated. Each feature's implementation will be explored, alongside the integration of AI models, their communication protocols, and the linkage of the Google Calendar API.

5.2 System Implementation

Our system implementation is divided into two main sections: backend (models integration) and frontend (web application development), each serving a distinct primary function. The backend focuses on utilizing Python for creating and integrating generative AI models and managing database interactions with SQLite. The web application development uses Flask to create a robust backend, while the frontend is built using HTML, CSS, and JavaScript to ensure a responsive and user-friendly interface.

5.2.1 Database

5.2.1.1 Connection

First, we established a connection to the database using SQLite3 in Python. This connection enabled us to interact with the database and perform various operations such as creating tables, inserting data, querying data, and updating records. Upon successful connection, we obtain a connection object that serves as our gateway to the database. This connection object allows us to execute SQL commands through a cursor, which acts as our interface to the database.

```python
import sqlite3

# Create or connect to the database
conn = sqlite3.connect('Mosaweeq.db')
print("connected to database successfully")
cursor = conn.cursor()
```

Figure 5.1: Database Connection

5.2.1.2 Tables Creation

In the database setup process, we created seven tables within the SQLite database as shown in Figure (Table List). These tables are essential elements of the system's data structure, each designed to store specific entities and their corresponding attributes.
As shown in Figure (Create Table), it how to create tables in database using Python.

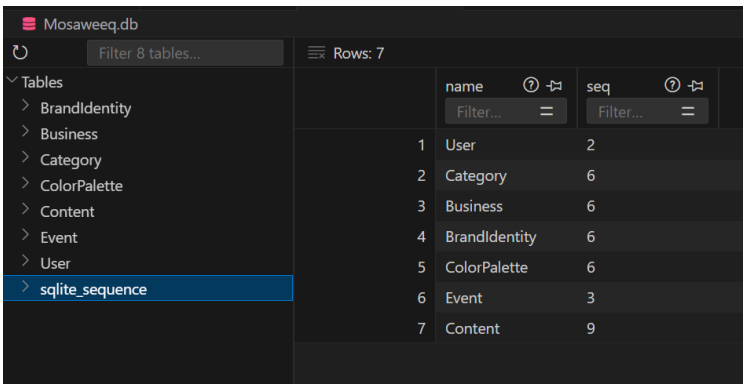


Figure 5.2: Table List

```
# Create User table
cursor.execute('''CREATE TABLE IF NOT EXISTS User (
    User_ID INTEGER PRIMARY KEY AUTOINCREMENT,
    Name VARCHAR(100) NOT NULL,
    Email VARCHAR(100) NOT NULL,
    Phone_Number VARCHAR(100),
    Password VARCHAR(255) NOT NULL,
    Country VARCHAR(100)
)''')
print("Table User created successfully")

# Create Business table
cursor.execute('''CREATE TABLE IF NOT EXISTS Business (
    Business_ID INTEGER PRIMARY KEY AUTOINCREMENT,
    User_ID INTEGER,
    Business_description TEXT,
    FOREIGN KEY (User_ID) REFERENCES User(User_ID)
)''')
print("Table Business created successfully")

# Create BrandIdentity table
cursor.execute('''CREATE TABLE IF NOT EXISTS BrandIdentity (
    Brand_ID INTEGER PRIMARY KEY AUTOINCREMENT,
    Brand_name VARCHAR(255),
    Logo_url VARCHAR(255),
    Business_ID INTEGER,
    FOREIGN KEY (Business_ID) REFERENCES Business(Business_ID)
)''')
print("Table BrandIdentity created successfully")
```

Figure 5.3: Create Table

5.2.2 Backend Implementation Using Flask and Python

The backend of our system was built using Python, which we used for both database interaction and AI model integration. Here is an overview of how Python and Flask were employed to create an efficient and scalable backend architecture:

Setting Up Routes: Flask routes define the endpoints of the application. Each route corresponds to a specific URL path and HTTP method (GET, POST, etc.), triggering the appropriate Python functions. This structure ensures clear and organized handling of HTTP requests and responses.

Integrating AI Models: Python functions, linked to Flask routes, handle the processing of data inputs through the generative AI models. This setup allows for dynamic content generation based on user inputs, providing real-time AI-driven responses and outputs.

Database Management: SQLite was used for the database. Python's seamless integration with SQLite allowed for efficient data manipulation and management operations, including inserting and updating data. This combination ensures reliable and efficient data storage and retrieval for user sessions and model outputs.
By using Flask and Python, the backend efficiently manages AI model integration, database interactions, and HTTP request processing. This approach creates a robust and scalable architecture, capable of supporting the dynamic needs of the application. The following section will discuss the backend implementation and AI models integration, beginning with the text generation model.

5.2.2.1 AI Model Configuration

5.2.2.1.1 Text Generation Model (Gemini)

In our system, we needed to use a text generation model to generate brand names, marketing taglines and descriptions of the upcoming events for the user.
The AI model configuration section of our code initializes and configures a generative AI model using Google's generative AI library. After importing the necessary library (google.generativeai), we configure the generative AI by providing an API key for authentication. The generation_config dictionary sets parameters for text generation, including temperature, top_p, top_k, and maximum output tokens, which control the creativity and scope of the generated content. Additionally, we define safety_settings to enforce content moderation, ensuring that generated outputs avoid harassment, hate speech, sexually explicit material, and dangerous content by blocking medium and above thresholds in these categories. Finally, we instantiate the generative AI model, specifying the model's name ("gemini-1.5-pro-latest"), and apply the defined configuration and safety settings. This setup ensures that our generative AI model produces safe, controlled, and relevant outputs for the application. The following code shows the configuration settings to set up the google generative AI Api.

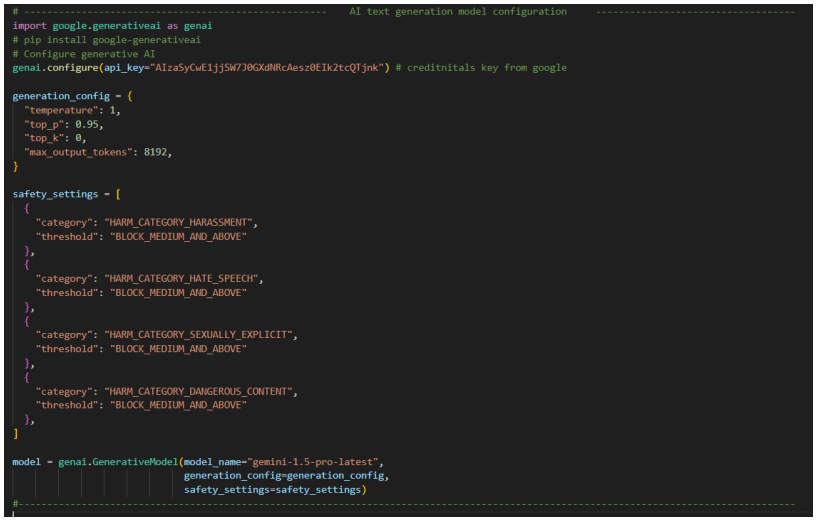


Figure 5.4: configuration settings

5.2.2.1.2 Image Generation Model

For the core of our system image generation model to generate posts and brand logo, we've implemented a configuration that interacts with multiple AI models hosted on Hugging Face's inference API. Our main goal is to generate images and logos, ensuring reliability with a fallback mechanism in case of failures.

1. API Endpoints and Headers:
We set up several endpoints and a hierarchical querying mechanism to manage potential failures and maintain service continuity. Starting with the configuration of API endpoints and headers for authentication, we proceed to implement functions for querying these endpoints.

Figure 5.5: API configuration

2. Queries Function:
In our approach, the first function 'queryLogo' targets a specific API for logo generation. This function handles the API request within a try-except block, ensuring that any HTTP-related errors are caught and logged.

Figure 0.6: QueryLogo function

The process of image generation initiates with the 'query' function, serving as the central point of contact for accessing image generation models. It starts by initiating a POST request to the primary image generation API (Image_API_URL), utilizing

predefined headers and payload data. If a failure occurs, indicated by a 'RequestException', signaling an inability to communicate with the primary model, we activate a fallback mechanism. This directs the process to the 'query_backup_model' function with the same payload. If no error is raised, indicating a successful response from the primary model, the function returns the generated image.

```python
def query(payload):
        try:
            response = requests.post(Image_API_URL, headers=headers, json=payload)
            response.raise_for_status()  # Raises an HTTPError for bad status codes
            return response.content
        except requests.exceptions.RequestException as e:
            logging.error("Error occurred with the primary model: %s", e)
            # Call a function to query the backup model
            return query_backup_model(payload)
```

Figure 0.7: query function

The following fallback functions (`query_backup_model`, `query_backup_model2`, query_backup_model3`) serve as alternative pathways in case the primary image generation API fails to respond. Each of these functions attempts to access a different image generation model hosted at distinct endpoints. If one backup model encounters an issue, the process seamlessly transitions to the next in line until a successful response is achieved or all available models are exhausted.

This multi-layered fallback strategy is essential for our applications that require high availability and reliability in image generation. By cascading through multiple backup endpoints, the system minimizes downtime and ensures continuous operation, even if one or more APIs become temporarily unavailable. This approach highlights the importance of redundancy and error handling in API-dependent services, enhancing overall system resilience.

```python
def query_backup_model(payload):
        try:
            response = requests.post(API_URL, headers=headers, json=payload)
            response.raise_for_status()  # Raises an HTTPError for bad status codes
            return response.content
        except requests.exceptions.RequestException as e:
            logging.error("Error occurred with the secondary model: %s", e)
            # Call a function to query the backup model
            return query_backup_model2(payload)

def query_backup_model2(payload):
        try:
            response = requests.post(BACKUP_API_URL, headers=headers, json=payload)
            response.raise_for_status()  # Raises an HTTPError for bad status codes
            return response.content
        except requests.exceptions.RequestException as e:
            logging.error("Error occurred with the tertiary model: %s", e)
            # Call a function to query the backup model
            return query_backup_model3(payload)

def query_backup_model3(payload):
        try:
            response = requests.post(Image_BACKUP_API_URL, headers=headers, json=payload)
            response.raise_for_status()  # Raises an HTTPError for bad status codes
            return response.content
        except requests.exceptions.RequestException as e:
            logging.error("Error occurred with the quaternary model: %s", e)
```

Figure 0.8: fallback functions

5.2.3 Frontend Implementation tools and languages

HTML, CSS, and JavaScript: HTML (Hypertext Markup Language), CSS (Cascading Style Sheets), and JavaScript were used to build frontend development.

5.2.3.1 Header:

The header section of the webpage contains a navigation bar and user authentication options. If the user is logged in, it displays a "Sign Out" button and the user's profile picture and name, linking to their profile page. If not logged in, it shows "Login" and "Sign Up" buttons. The navigation bar includes links to sections like "Contact Us," "Our Services," and "About Us," along with a link to the homepage. Additionally, it features the application's logo. Overall, it provides easy access to essential features and information. As shown in Figure (Header HTML):

```
<header class="header">
    <div class="container">
        <div class="left-section">
            {% if user_id %}
            <!-- Display user information if user is logged in -->
            <a href="{{ url_for('logout') }}" onclick="return confirmLogout()"><button id="signOut">تسجيل
                    الخروج</button></a>
            <a href="{{ url_for('profile') }}" class="profile-info" id="profile">
                <img src="{{ url_for('static', filename='images/profileIcon.jpg') }}" class="userIconn"
                    alt="Profile Icon">
                <span class="username">{{ full_name }}</span>
            </a>
            {% else %}
            <!-- Display login/signup buttons if user is not logged in -->
            <form action="/login" method="POST">
                <button id="login">تسجيل دخول</button>
            </form>
            <form action="/signup" method="GET">
                <button id="signup">انشاء حساب</button>
            </form>
            {% endif %}
        </div>
        <div class="menu-btn" onclick="toggleNav()">☰</div>
        <nav>
            <li><a href="#contact">تواصل معنا</a></li>
            <li><a href="#Services">خدماتنا</a></li>
            <li><a href="#HowIsMosaweeq?">من نحن</a></li>
            <li><a href="{{ url_for('home') }}">الصفحة الرئيسية</a></li>
        </nav>
        <div class="Headerlogo">
            <img src="{{ url_for('static', filename='images/Mosaweeq.png') }}" alt="Mosaweeq Logo">
        </div>
    </div>
</header>
```

Figure 0.9: Header HTML

It checks if the user is logged in by verifying their user ID in the session. If the user is logged in, it renders the homepage with their user ID and full name. If not logged in, it redirects the user to the homepage. Additionally, it clears any event data from the session to ensure a refreshed state. As shown in Figure (User_Id Session):

```
@app.route("/")
def home():
    #Check if the user is logged in by checking if their user ID is in the session
    user_id = session.get("user_id")
    full_name = session.get("full_name")

    # delete events from session to refresh if there is a session
    if 'event_name' in session:
        del session['event_name']

    if user_id:
        # If user is logged in, render the home page with their user ID
        return render_template("Homepage.html", user_id=user_id,full_name=full_name)
    else:
        # If user is not logged in, redirect them to the Homepage
        return render_template("Homepage.html")
```

Figure 0.10: User_Id Session

5.2.3.2 Sign Up Page:

Users can create their own account by filling the sign-up form as shown in the Figure (Sign Up in HTML Form):

```
<div class="col right-section">
    <form class="sign-up" method="POST" action="/signup" onsubmit="return validatePassword()">
        <div class="clearfix">
            <h2 class="heading mb-4">انضم الى مُسوّق</h2>
        </div>
        <div class="form-row">
            <div class="form-group2 fone">
                <label>
                    <p>الأسم الأخير</p>
                    <input type="text" name="first_name" class="form-control" />
                </label>
            </div>
            <div class="form-group1 fone">
                <label>
                    <p>الأسم الأول</p>
                    <input type="text" name="last_name" class="form-control" />
                </label>
            </div>
        </div>
        <div class="form-row">
            <div class="form-group1 fone">
                <label>
                    <p>رقم الجوال</p>
                    <input type="text" name="phone_number" class="form-control" placeholder="05XXXXXXXX" />
                </label>
            </div>
            <div class="form-group2 fone">
                <label>
                    <p>الإيميل</p>
                    <input type="email" name="email" class="form-control" placeholder="XXXXX@gmail.com" />
                </label>

            </div>
```

Figure 0.11: Sign Up in HTML Form

```
    <div class="form-row">
        <div class="countryStyle ">
            <label>
                <p>الدولة: </p>
                <select name="country" id="countrySelect">
                    <option value="" >اختر الدولة</option>
                    <option value="السعودية">السعودية</option>
                    <option value="مصر">مصر</option>
                    <option value="الإمارات">الإمارات</option>
                    <option value="البحرين">البحرين</option>
                    <option value="الجزائر">الجزائر</option>
                    <option value="العراق">العراق</option>
                    <option value="الأردن">الأردن</option>
                    <option value="لبنان">لبنان</option>
                    <option value="ليبيا">ليبيا</option>
                    <option value="المغرب">المغرب</option>
                    <option value="فلسطين">فلسطين</option>
                    <option value="قطر">قطر</option>
                    <option value="سوريا">سوريا</option>
                    <option value="تونس">تونس</option>
                    <option value="اليمن">اليمن</option>
                    <option value="الكويت">الكويت</option>
                    <!-- You can continue adding more countries as needed -->
                </select>
            </label>
        </div>
    </div>
    <div class="form-row">
        <div class="form-group2 fone">
            <label>
                <p>تأكيد كلمة المرور</p>
                <input type="password" name="confirm_password" id="confirm_password"
                    class="form-control" placeholder="**********"/>
            </label>
        </div>
        <div class="form-group1 fone">
            <label>
                <p> كلمة المرور</p>
                <input type="password" name="password" id="password" class="form-control"
                    placeholder="**********" />
            </label>
        </div>
    </div>
```

Figure 0.12: Sign Up in HTML Form

The processing of the user request must go through a set of conditions to validate his input before creating a record in the database. The first step is to validate his email and search

on the database to make sure this email is new and does not exist before, as shown in Figure (Validate Email)

```python
def check_email_existence(email):
    try:
        con = sqlite3.connect('Mosaweeq.db')
        cur = con.cursor()
        cur.execute("SELECT * FROM User WHERE Email=?", (email,))
        existing_user = cur.fetchone()
        return existing_user is not None
    except sqlite3.Error as e:
        print("Error:", e)
        return False
    finally:
        if con:
            con.close()
```

Figure 0.13: Validate Email

Second step is to check if the password contains at least eight characters, including at least one digit and one special character from the defined set of special characters, as shown in Figure (Validate Password)

```python
# Regular expression pattern for password validation
password_pattern = r"^(?=.*\d)(?=.*[!@#$%^&*])[a-zA-Z0-9!@#$%^&*]{8,}$"

def validate_password(password):
    """
    Validate the password based on defined criteria.
    Returns True if the password is valid, otherwise False.
    """
    return bool(re.match(password_pattern, password))
```

Figure 0.14: Validate Password

The JavaScript function `validatePassword()` compares the password and confirms password fields to ensure they match during signup. If they don't match, an error message is displayed, as shown in Figure (Confirm Password)

```javascript
<script>
    function validatePassword() {
        var password = document.getElementById("password").value;
        var confirmPassword = document.getElementById("confirm_password").value;
        var errorMessage = document.getElementById("password_error");



        if (password !== confirmPassword) {
            errorMessage.innerText = "كلمة المرور وتأكيد الرقم السري غير متطابقين.";
            errorMessage.style.display = "block";
            errorMessage.style.direction="rtl";

            return false;
        } else {
            errorMessage.style.display = "none";
        }

        return true;
    }
</script>
```

Figure 5.15: Confirm Password

If the data entered has passed all the previous conditions, then a new record will be created, as shown in Figure (insert in User Table).

```python
@app.route("/signup", methods=["POST"])
def signUp():
    con = None  # Initialize con variable
    try:
        if request.method == "POST":
            # Get form data
            form_data = request.form
            required_keys = ["first_name", "last_name", "email", "phone_number", "password", "country"]
            if all(key in form_data for key in required_keys):
                first_name = form_data["first_name"]
                last_name = form_data["last_name"]
                email = form_data["email"]
                phone_number = form_data["phone_number"]
                password = form_data["password"]
                country = form_data["country"]

                # Password validation
                if not validate_password(password):
                    error_msg = "كلمة المرور يجب أن يتكون من 8 حروف على الأقل ويحتوي على رقم وحرف خاص على الأقل."
                    return render_template("signup.html", error_pas=error_msg)

                # Connect to SQLite3 database and execute the insertion
                con = sqlite3.connect('Mosaweeq.db')  # Assign value to con
                cur = con.cursor()

                # Check if email already exists in the database
                if check_email_existence(email):
                    error_msg = "الإيميل موجود مسبقًا."
                    return render_template("signup.html", error=error_msg)
```

Figure 5.16: insert in User Table

Before proceeding with creating an account, the user is prompted to indicate whether they possess a business identity for their project as shown in figure (have brand Identity for your business or not).
When the user selects "Yes," they're prompted to provide detailed business identity information, including category selection, company name and description, logo upload, and primary colors associated with their business identity. If "No" is chosen, the process is simplified, allowing direct account creation without business identity input.

```html
    <p class="contentq"><b>هل لديك هوية لشركتك؟</b> (شعار لشركتك واسم)</p>
    <div class="form-row">
        <div class="form-group1">
            <button type="button" name="no" class="btns">لا</button>
        </div>
        <div class="form-group2">
            <button type="button" name="yes" class="btns">نعم</button>
        </div>
    </div>
    <div class="comInfo"></div>
</form>
```

Figure 0.17: have brand Identity for your business or not

Following the user's "YES" choice, detailed business identity information, such as category selection, company name, description, logo upload, and and primary colors associated with their business identity, is collected and inserted into Tables., as shown in Figure (insert business identity information).

```python
# Set the user ID in the session after signing up
session['user_id'] = user_id
session['full_name'] = first_name + " " + last_name

# Check if the user selected "Yes" for company info
if "companyName" in form_data and form_data["companyName"]:
    # Insert company information into the Business table
    business_description = form_data.get("companyDescription", "")
    cur.execute("INSERT INTO Business (User_ID, Business_description) VALUES (?, ?)",
                (user_id, business_description))
    business_id = cur.lastrowid  # Retrieve the numeric business ID from the database

    # Insert brand identity into the BrandIdentity table
    brand_name = form_data.get("companyName", "")
    logo_url = form_data.get("logo", "")  # Assuming you handle logo upload elsewhere
    cur.execute("INSERT INTO BrandIdentity (Brand_name, Logo_url, Business_ID) VALUES (?, ?, ?)",
                (brand_name, logo_url, business_id))

    # Insert category into the Category table
    category = form_data.get("category", "")
    cur.execute("INSERT INTO Category (Category_Name, Business_ID) VALUES (?, ?)",
                (category, business_id))

    # Insert colors into the ColorPalette table
    color1 = form_data.get("color1", "")
    color2 = form_data.get("color2", "")
    color3 = form_data.get("color3", "")
    cur.execute("INSERT INTO ColorPalette (Color1, Color2, Color3, Brand_ID) VALUES (?, ?, ?, ?)",
                (color1, color2, color3, business_id))

    # Insert other company-related information here

con.commit()  # Commit the transaction
```

Figure 5.18: insert business identity information

5.2.3.3 Sign In Page:

In order to login to the system the user need to fill the form in the login page as shown in Figure (Log in HTML Form):

```html
<div class="container">
    <div class="row">
        <div class="col left-section">
            <div class="img-overlay"></div>
        </div>
        <div class="col right-section">
            <form class="sign-up" method="POST" action="/login">
                <div class="clearfix">
                    <h2 class="heading mb-4">الدخول الى مُصوّق</h2>
                </div>
                <!-- Print error message here -->
                <div class="error-message" id="error-message"></div>
                <div class="form-row">
                    <div class="form-group2 fone">
                        <label>
                            <p>الإيميل</p>
                            <input type="email" name="email" id="email" class="form-control"
                                placeholder="XXXXX@gmail.com" />
                        </label>
                    </div>
                </div>
                <div class="form-row">
                    <div class="form-group1 fone">
                        <label>
                            <p>الرقم السري</p>
                            <input type="password" name="password" id="password" class="form-control"
                                placeholder="6 حروف ف أكثر" />
                        </label>
                        <div class="image"><i class="fas fa-eye"></i></div>
                    </div>
                </div>
                <div class="form-group2">
                    <button type="submit" name="signin" id="signin" class="btns">تسجيل الدخول</button>
                </div>
                <p class="forget"> هل نسيت كلمة المرور؟ <a class="forgetaa"> اضغط هنا</a></p>
            </form>
        </div>
    </div>
</div>
```

Figure 0.19: Log in HTML Form

The provided code defines a Flask route "/login" to handle user authentication. Upon receiving POST requests with email and password data, it queries the SQLite database to find a matching user. If found, it compares the provided password with the stored password. Successful logins initiate a session and redirect users to the homepage, displaying their full name. Invalid passwords prompt error messages, enhancing security and user feedback, as shown in Figure (Login HTML Form):

```python
# Route to render the login page
@app.route("/login")
def loginPage():
    return render_template("login.html")

# Route to handle login authentication
@app.route('/login', methods=['POST'])
def login():
    # Get form data
    email = request.form.get('email')
    password = request.form.get('password')

    # Connect to database
    con = sqlite3.connect('Mosaweeq.db')
    cursor = con.cursor()
    cursor.execute("SELECT * FROM user WHERE email = ?", (email,))
    user = cursor.fetchone()

    if user:
        if user[4] == password:  # Assuming password is stored in the fifth column
            # Successful login
            session['user_id'] = user[0]  # Use the user ID from the database
            session['full_name'] = user[1]
            user_id = session.get("user_id")
            print(session['full_name'])
            return render_template("Homepage.html", full_name=session['full_name'], user_id=user_id)
        else:
            # Invalid password
            return render_template("login.html", message="كلمة المرور غير صحيحة. الرجاء المحاولة مرة أخرى.")

    # If user is not found or password is incorrect
    return render_template("login.html", message="البريد الإلكتروني غير موجود أو كلمة المرور غير صحيحة. الرجاء المحاولة مرة أخرى.")
```

Figure 5.20: Login HTML Form

5.2.3.4 Logout:

The `/logout` route clears all session variables when a user logs out, effectively ending their session. After clearing the session, it redirects the user to the homepage, as shown in Figure (Logout):

```python
@app.route('/logout')
def logout():
    # Clear all session variables
    session.clear()
    # Redirect to the home
    return redirect(url_for('home'))
```

Figure 0.21: Logout

The `confirmLogout()` JavaScript function presents users with a confirmation message in Arabic, ensuring clarity in communication. This function serves as a safeguard against inadvertent logouts, enhancing user experience by offering a deliberate choice before proceeding with the logout action. If users confirm their intention to log out, the function closes the session and initiates the logout process, ensuring secure and controlled access to the platform, as shown in Figure (Confirm LogOut):

```javascript
<script>
    function confirmLogout() {
        // Display a confirmation dialog
        var confirmMsg = "هل أنت متأكد من تسجيل الخروج؟ سيتم إغلاق الجلسة وتسجيل الخروج";
        return confirm(confirmMsg);
    }
</script>
```

Figure 0.22: Confirm LogOut

5.2.3.5 Home Page:

The Home page serves as the main hub for users and visitors. It features a header with navigation links and user authentication options. The main content highlights the application's focus on simplifying marketing through artificial intelligence. Users can explore various services offered, such as creating brand identity and generating marketing content. The page also includes a contact section for user feedback. In the footer, users

find navigation links, social media links, and a copyright notice. Overall, it's a concise and engaging introduction to the application's features and functionalities.as shown in Figure (HomePage HTML)

```
<div class="main-content">
    <div class="tittle">
        <span class="title">خلي التصويق علي </span>
        <span class="title1">مُسوّق</span>
    </div>
    <div class="para">
        <span class="first">استفد من قوة <b>الذكاء الاصطناعي</b>في حملات التسويق الخاصة بك</span>
        <span class="second">الى فريق تسويق مخصص أو خبره سابقة <b>دون الحاجة</b></span>
    </div>
    <div class="divServsButt">
        <a href="{{ url_for('brandingPage') }}">
            <button>
                <img src="{{ url_for('static', filename='images/Mask group (2).png') }}" alt="Your Image" />
                <p class="create-content-button">انشئ هويتك التجارية</p>
            </button>
        </a>

        <a href="{{ url_for('generateContentWithEvent') }}">
            <button>
                <img src="{{ url_for('static', filename='images/Mask group (1).png') }}" alt="Your Image" />
                <p class="create-content-button">انشئ محتوى لمناسبة القادمة</p>
            </button>
        </a>

        <a href="{{ url_for('generateContent') }}">
            <button>
                <img src="{{ url_for('static', filename='images/Mask group.png') }}" alt="Your Image" />
                <p class="create-content-button">انشئ محتوى تسويقي جديد</p>
            </button>
        </a>

    </div>
    <h1 id="HowIsMosaweeq?" class="question">من هو مسوق؟</h1>
    <div class="definition">
        <span class="definition1">هو تطبيق ويب يجعل التسويق بسيطًا وفعالًا تمكن المنصة الشركات والأفراد من
            إنشاء محتوى تسويقي جذاب باللغة العربية يتصل بجمهورهم. </span>
        <span class="definition2">نقوم بدمج تكنولوجيا الذكاء الاصطناعي التوليدية بشكل سلس مع إدعاءات العلامة
            التجارية لتبسيط توسيع الأعمال لأصحاب الأعمال والمصمم لهم بالتركيز فقط على جودة منتجاتهم أو خدماتهم
        </span>
    </div>
    <h1 id="Services" class="ourServices">خدماتنا</h1>
    <div class="row">
        <div class="col">
            <h3>انشاء اسم للهوية التجارية</h3>
            <p>تساعد الخدمة في اختيار وتكوين اسم قوي وذي طابع فريد يتناسب مع هوية العلامة التجارية ويترك
                انطباعًا قويًا لدى الجمهور المستهدف.</p>
        </div>
        <div class="col">
            <h3>انشاء شعار للهوية التجارية</h3>
            <p>تقدم الخدمة إمكانيات فائقة في إنشاء شعارات فريدة ومبتكرة تعبر بوضوح عن هوية العلامة التجارية. يتم
                التركيز على تصميم جذاب وذكي يتناغم مع رؤية العميل ويساهم في تميزهم في سوق المنافسة.</p>
        </div>
        <div class="col">
            <h3>انشاء محتوى لمناسبة قادمة </h3>
            <p>تقدم خدمتنا حلاً مبتكرًا وفعالًا للشركات الراغبة في تحسين استراتيجياتها التسويقية عبر مزامنة
                المحتوى مع الفعاليات والعطل الرسمية. يعد هذا النهج استراتيجيًا حيويًا للتواصل الفعّال مع الجمهور
                في الأوقات المثلى.</p>
        </div>
```

Figure 0.23: HomePage HTML

5.2.3.6 Profile Page:

The profile page displays the user's information, as shown in Figure (Profile HTML Form) and enables the user to modify their information.

```
<div class="row">
    <div class="main-content">
        <div id="section1" class="content-section active" style="display: block;">
                <!-- Display the update message if any -->
                {% if update_message %}
                <div class="error-message" style="color:green;">{{ update_message }}</div>
                <script>
                    // Remove the update message after 5 seconds
                    setTimeout(function() {
                        document.querySelector('.error-message').style.display = 'none';
                    }, 3000);
                </script>
                {% endif %}
                <!-- Display the form with user data -->
                <form id="updateForm" class="rectanglee" method="post" action="">
                    <div class="text">
                        <p>الاسم :</p>
                        <input type="text" id="firstName" name="firstName" placeholder=" الاسم " value="{{ first_name }}" required>
                    </div>
                    <div class="text">
                        <p>رقم الجوال :</p>
                        <input type="text" id="phoneNumber" name="phoneNumber" placeholder=" 05********" value="{{ phone_number }}" required>
                    </div>
                    <div class="text">
                        <p>الايميل :</p>
                        <input type="text" id="email" name="email" placeholder=" الايميل" value="{{ email }}" required>
                    </div>
                    <div class="text">
                        <p>كلمة المرور :</p>
                        <input type="text" id="password" name="password" placeholder=" +characters" value="{{ password }}" required>
                    </div>
                    <div class="text">
                        <button class="saveButton" type="submit" name="submit">حفظ التغيرات</button>
                    </div>
                </form>
        </div>
</div>
```

Figure 0.24: Profile HTML Form

This route ("/profile") renders the user's profile page. It first checks if the user is logged in by verifying the presence of their user ID in the session. If the user is not logged in, they are redirected to the login page.
After confirming the user's login status, the route retrieves the user's information from the database based on their user ID. It fetches data such as first name, email, phone number, and password, as shown in Figure (retrieves the user information)
Finally, it renders the "Profile.html" template, passing the retrieved user data as context variables to be displayed on the profile page.

```python
# Route to render the profile page
@app.route("/profile")
def profile():
    if 'user_id' not in session:
        return redirect(url_for('loginPage'))

    user_id = session['user_id']
    conn = get_db_connection()
    cursor = conn.cursor()

    # Fetch user data
    cursor.execute("SELECT * FROM User WHERE User_ID = ?", (user_id,))
    user = cursor.fetchone()

    conn.close()

    if user:
        # If user is found, fetch the user data
        first_name = user['Name']
        email = user['Email']
        phone_number = user['Phone_Number']
        password = user['Password']
    else:
        return "User not found"

    return render_template('Profile.html', first_name=first_name, email=email, phone_number=phone_number,
                           password=password)
```

Figure 5.25: retrieves the user information

This route handles the updating of a user's profile information. It checks if the user is logged in, collects the updated information from the form, updates the user's data in the database, and stores the new full name in the session. Finally, it renders the profile page with the updated information and a confirmation message, as shown in Figure (Update the user information)

```python
# Route to handle updating user profile
@app.route("/profile", methods=['POST'])
def update_profile():
    if 'user_id' not in session:
        return redirect(url_for('loginPage'))

    user_id = session['user_id']

    if request.method == 'POST':
        # Collect form data
        first_name = request.form['firstName']
        phone_number = request.form['phoneNumber']
        email = request.form['email']
        password = request.form['password']

        conn = get_db_connection()
        cursor = conn.cursor()

        # Prepare and execute SQL statement to update user information
        cursor.execute("UPDATE User SET Name=?, Phone_Number=?, Email=?, Password=? WHERE User_ID = ?",
                       (first_name, phone_number, email, password, user_id))

        conn.commit()
        # if the name was updated, we have to update it in the session too
        cursor.execute("SELECT User_ID, Name FROM User WHERE User_ID = ?", (user_id,))
        conn.commit()
        username = cursor.fetchone()
        session['full_name']=username[1] # since column 1 contain the full name

        # Close the connection
        conn.close()
        update_message = "لقد تم حفظ التغييرات"

    return render_template('Profile.html', first_name=first_name, email=email, phone_number=phone_number,
                           password=password, update_message=update_message)
```

Figure 0.26: Update the user information

5.2.3.7 Brand Identity Page:

The first service available in MOSAWEEQ is creating a Brand identity.

Brand Identity Section 1:
First, the user provides information about the business by selecting the business category. Then the user chooses whether the name he wants to create by the form be in Arabic or English. Finally, the user writes a description of the business, as shown in Figure (Brand identity section1 HTML Form)

```
<div class="row">
    <div class="main-content">
        <div class="w3-container w3-teal">
            <h3 style="direction: rtl ;color:white ;padding-bottom: 10px;">انشئ هويتك التجارية</h3>
        </div>
        <!-- Content Sections -->
        <div class="content-section active">
            <form class="rectanglee" action="/branding/step1" method="POST">
                <div class="text">
                    <p>اختر مجال شركتك:</p>
                    <select required name="category">
                        <option value="" disabled>المجالات</option>
                        <option value="ازياء">ازياء</option>
                        <option value="الاطعمه ومشروبات">الاطعمه ومشروبات</option>
                        <option value="العطور">العطور</option>
                        <option value="المجوهرات">المجوهرات</option>
                        <option value="الخدمات">الخدمات</option>
                        <option value="الجمال">الجمال</option>
                        <option value="الفنون">الفنون</option>
                        <option value="الصحة واللياقة البدنية">الصحة واللياقة البدنية</option>
                    </select>
                </div>
                <div class="text">
                    <p>كتابة الاسم باللغة: </p>
                    <div class="btn-group" role="group" aria-label="Basic radio toggle button group">
                        <input type="radio" class="btn-check" name="btnradio" id="btnradio1" autocomplete="off"
                            value="العربية">
                        <label class="btn btn-outline-primary" for="btnradio1">العربية</label>

                        <input type="radio" class="btn-check" name="btnradio" id="btnradio2" autocomplete="off"
                            value="English">
                        <label class="btn btn-outline-primary" for="btnradio2">English</label>
                    </div>
                </div>
                <div class="text">
                    <p>وصف: </p>
                    <div>
                        <textarea id="description" placeholder="مثل...." name="companyDescription" required></textarea>
                    </div>
                </div>
                <input class="SaveButtoon" type="submit" name="info" value="التالي">
            </form>
        </div>
    </div>
```

Figure 0.27: Brand identity section1 HTML Form

The provided code handles the branding page's functionality for creating a brand identity in MOSAWEEQ. When a user accesses the `/branding/step1` route, the form collects business information, including the business category, description, and preferred language for the brand name. If the user submits the form (via POST request), the data is stored in the session. An input text is then structured and sent to an AI model to generate six brand name suggestions in the specified language. The results are processed, cleaned, and then displayed on a new page (`brand-section2.html`). If the request is a GET request, the initial branding form (`branding1.html`) is rendered. If the user is not signed in, an error message is displayed, as shown in Figure (Store step1 in Session)

```python
# Route to render the branding page
@app.route('/branding/step1', methods=['GET', 'POST'])
def branding_step1():
    if request.method == "POST":
        # Get form data
        category = request.form.get('category')
        companyDescription = request.form.get('companyDescription')
        language = request.form.get('btnradio')  # Retrieve selected language
        # Retrieve user ID from session
        if 'user_id' in session:
            user_id = session['user_id']

            # Store data in session
            session['step1_data'] = {'category': category, 'companyDescription': companyDescription}
```

Figure 0.28: Store step1 in Session

After obtaining a short description of the brand, its category, and the preferred language for the brand name from the user, we designed a prompt to send to the AI model to generate brand name suggestions. Below is the prompt used:

```
# in this part we structure the promt sent to the AI model
input_text = f"اقترح لي 6 أسماء باللغة {language} لعلامتي التجارية التي تتخصص في مجال
{category} والتي تقدم {companyDescription} "
```

Figure 0.29: Brand name prompt

The following code snippet illustrates how our system interacts with the AI model to generate responses based on the prompt:

```
convo = model.start_chat(history=[])
convo.send_message(input_text)
# Split the text into lines
output = convo.last.text.split('\n')
```

Figure 0.30: System Interaction with AI Model

In this code, we initiate a new chat session with the AI model using convo = model.start_chat(history=[]).
Next, we send the designed prompt with convo.send_message(prompt). Finally, we retrieve the model's response using output = convo.last.text.split('\n'), which splits the response into a list of lines for easier processing.
The following code snippet processes the output from the AI model to extract and clean brand names. First, an empty list called brandNames is initialized to store the processed names. The code then checks if the output contains at least two lines, ensuring there's a headline followed by at least one brand name. It iterates over each line starting from the second line (skipping the headline). Each line is split by the delimiter '. ', and the second part of the split (the brand name) is extracted, stripped of leading and trailing whitespace, and any asterisks are removed. The cleaned brand name is then appended to the brandNames list. This ensures the final list contains well-formatted brand names ready for display.

```
# Initialize an empty list to store processed brand names
brandNames = []

# Check if there are at least two lines (headline + at least one brand name)
if len(output) >= 2:
    # Iterate over lines starting from the second line (skipping the headline)
    for line in output[1:]:
        # Split the line by '. ' and take the second part (after the number)
        parts = line.split('. ')
        if len(parts) > 1:  # Ensure there is content after the split
            name = parts[1].strip()  # Remove leading and trailing whitespace
            name = name.replace('**', '')
            brandNames.append(name)  # Add the processed brand name to the list
```

Figure 0.31: Brand Name Extraction and Cleaning Process

lastly, we will send the brand names to the next rendered page to be displayed.

```
    # Pass the AI result to the template for step 2
    return render_template('brand-section2.html', brandNames=brandNames)
else:
    return("you are not signed in")
```

Figure 0.32: Sending Brand Names to Rendering Page

Brand Identity Section 2:

The HTML code represents the second step in the brand identity creation process on the MOSAWEEQ platform. In this step, the user selects a brand name from a list of suggestions generated in the previous step. Each suggested brand name is presented as a radio button option, ensuring that the user must select one before proceeding to the next step, as shown in Figure (Brand identity section 2 HTML Form)

```
<div class="row">
    <div class="main-content">
        <div class="w3-container w3-teal ">
            <h3 style="direction: rtl ;color: white ;padding-bottom: 10px;">انشئ هويتك التجارية</h3>
        </div>

        <div class="content-section active">
            <form  class="rectanglee" id="brandSelectionForm" method="POST" action="/branding/step2">
                <div role="group" aria-label="Basic radio toggle button group">
                    <!-- Iterate over the brandNames and display each as a radio button option -->
                    {% if brandNames %}
                    {% for name in brandNames %}
                    <span class="CompanyName-group">
                        <input type="radio" class="CompanyName-check" name="CompanyName"
                            id="CompanyName{{ loop.index }}" value="{{ name }}" autocomplete="off" required>
                        <label class="CompanyName btn-outline-primary" for="CompanyName{{ loop.index }}">{{ name
                            }}</label>
                    </span>
                    {% endfor %}
                    {% endif %}
                </div>
                <!-- Next Button -->
                <div class="w3-container w3-teal MoveButton">
                    <!-- Use JavaScript to validate selection before submission -->
                    <input type="submit" class="next" value="التالي" onclick="validateSelection(event)"></input>
                </div>
            </form>
        </div>
    </div>
```

Figure 0.33: Brand identity section 2 HTML Form

The code defines the branding_step2 route for handling the second step in the brand identity creation process on the MOSAWEEQ platform. When the user submits the form via POST, it retrieves the selected company name from the form and the user ID from the session. The company name is cleaned by removing any additional meanings or descriptions, then stored in the session. The user is then directed to the next step by rendering the brand-section3.html template, as shown in Figure (Store step 2 in Session)

```python
@app.route('/branding/step2', methods=['GET', 'POST'])
def branding_step2():
    if request.method == 'POST':
        # Retrieve user ID from session
        if 'user_id' in session:
            user_id = session['user_id']

            # Get the selected company name
            company_name = request.form['CompanyName']

            # to remove the meaning from the name
            company_name = company_name.split(":", 1)[0].strip()

            # Save the selected company name in session data
            session['step2_data'] = {'company_name': company_name}

            # Render template for step 3
            return render_template('brand-section3.html')

        # If user ID is not found in session, handle the error
        return "User ID not found in session."

    # If it's a GET request, render the template for the second step
    return render_template('brand-section2.html')
```

Figure 0.34: Store step 2 in Session

Brand Identity Section 3:
The HTML code represents the third step in the brand identity creation process on the MOSAWEEQ platform. In this step, users are prompted to select colors for their brand. Once the user has made their selections, they can proceed to the next step by clicking the "Next" button, as shown in Figure (Brand identity section3 HTML Form)

```
<div class="row">
    <div class="main-content">
        <div class="w3-container w3-teal">
            <h3 style="direction: rtl ;color: white ;padding-bottom: 10px;">اضفي هويتك التجارية</h3>
        </div>
        <div id="section3" class="content-section active">
            <form class="rectanglee" method="POST" action="/branding/step3">
                <!-- Add content for section 3 here -->
                <div class="text">
                    <p>اختر الوان شركتك:</p>
                </div>
                <div class="divColor" data-tooltip="Select color">
                    <!-- Add color boxes with ids for storing the colors -->
                    <input type="hidden" id="color1" name="color1" value="">
                    <input type="hidden" id="color2" name="color2" value="">
                    <input type="hidden" id="color3" name="color3" value="">
                    <div class="color-box" data-tooltip="Select color"></div>
                    <div class="color-box" data-tooltip="Select color"></div>
                    <div class="color-box" data-tooltip="Select color"></div>
                </div>
                <div class="text">
                    <p> *يمكنك تغيير اللون عن طريق النقر عليه.</p>
                </div>

                <!-- Next Button -->
                <div class="w3-container w3-teal MoveButton">
                    <input type="submit" name="color" type="button" class="next" value="التالي"></input></a>
                </div>
            </form>
        </div>
    </div>
```

Figure 0.35: Brand identity section3 HTML Form

The `/branding/step3` route finalizes the branding process on the MOSAWEEQ platform. it retrieves the selected colors from the form and previously stored data from the session. This includes the company name, category, and description. The retrieved data is stored in `step3_data` in the session. This route prepares to send the data to the next page for review and ensures all gathered information is ready to be saved to the database, marking the completion of the branding process, as shown in Figure (Store step3 in Session)

```
@app.route('/branding/step3', methods=['GET', 'POST'])
def branding_step3():
    if request.method == 'POST':
        # Retrieve user ID from session
        if 'user_id' in session:
            user_id = session['user_id']

            color1 = request.form.get('color1')
            color2 = request.form.get('color2')
            color3 = request.form.get('color3')

            # Extract relevant data
            step2_data = session.get('step2_data', {})
            company_name = step2_data.get('company_name')
            brand_name = company_name

            # Store data in session
            session['step3_data'] = {'brand_name': brand_name, 'color1': color1, 'color2': color2, 'color3': color3}

            # we will send the result to the next page
```

Figure 0.36: Store step3 in Session

In the third step of the branding process, we utilize an AI model to retrieve the names of the colors chosen by the user, converting them from hexadecimal values to recognizable names.

```
colorname = "write the shade name of those colors, each color name start by ^^^^ :"+color1+" "+color2+" "+color3
colors = generate_advertisement(colorname)
```

Figure 0.37: Color Naming with AI: From Hex Codes to Recognizable Names

Then, we design a prompt containing essential information such as business description, category name, and colors, which is then sent to the logo generation model. The structure of this prompt includes specific placeholders for each piece of information, ensuring clarity and coherence in the generated text. As shown below:

```
description = "Translate the sentence to English 6 times in this structure of a sentence: (logo, A logo for " + \
  company_description + ", " + category_name + ", dynamic, energetic colors (" + colors + \
  "), LogoRedAF, Icons). Each translation should start with ^^^^."
```

Figure 0.38: Logo prompt

After designing the prompt structure, we send it to the text generation model to translate it into multiple variations of English sentences to introduce diversity. Each translation follows a specific structure describing the logo, brand, category, and color scheme. Following this, these translations go through a cleaning process to remove unnecessary characters and formatting, ensuring readability before being sent to the logo generation model.

```
translated_description = generate_advertisement(description)
translations = clean_translations(translated_description)
```

Figure 0.39: Diverse Sentence Generation & Streamlined Cleaning Processes

```
def clean_translations(text):
    # Remove leading and trailing whitespaces
    text = text.strip()
    # Remove unnecessary characters and empty lines
    cleaned_text = '\n'.join(line.strip() for line in text.split('\n') if line.strip())

    # Decode the cleaned text using UTF-8 encoding
    decoded_text = cleaned_text.encode('utf-8').decode('utf-8', 'ignore')

    cleaned_translations = []
    translation = decoded_text.splitlines()
    i = 0
    for line in translation:
        idx = line.find("^^^^")
        if idx != -1:  # If "^^" is found
            cleaned_line = line.replace("^^^^", "")  # Remove "^^^^"
            cleaned_translations.append(cleaned_line)

    return cleaned_translations
```

Figure 0.40: Cleaning Translations function

For each cleaned translation, an attempt is made to generate a corresponding logo image using the `generate_Logo` function. This function sends a prompt to the `queryLogo` logo generation model, which returns an image based on the provided text description. If any errors occur during image generation, the process seamlessly continues to the next translation.

Once a logo image is successfully generated, it is encoded into a base64 format for compatibility and stored in the `generated_logo` list.

```
if translations:
  for idx in range(len(translations)):
    # Get the translation at the current index
        translation = translations[idx]
        # Generate image based on the translated description
        try:
            generated_image = generate_Logo(translation)
        except ValueError as e:
        # Continue to the next iteration if an error occurs during image generation
            continue
        if generated_image is not None:
            # Encode the image to base64
            buffer = io.BytesIO()
            generated_image.save(buffer, format="JPEG")
            base64_encoded_image = base64.b64encode(buffer.getvalue()).decode()
            # Decode the base64 string back into an image
            generated_logo.append(base64_encoded_image)
        else:
            logging.error("Generated image is None.")
```

Figure 0.41: logo generation process

The `generate_Logo` function itself initiates a POST request to the `queryLogo` endpoint, passing along the prompt and additional parameters such as negative prompts, guidance scale, and temperature. Once the image data is received, the function checks the image format to ensure it is supported and valid. If the image format is unsupported or the data is corrupted, an appropriate error is raised.

```
def generate_Logo(prompt):
    image_bytes = queryLogo({
            "inputs": prompt,
            "negative_prompt": [
            "unprofessional",
            "uninspired",
            "generic",
            "unrealistic",
            "text",
            "abstract backgrounds",
            "dark backgrounds",
            "distorted",
            "blurred",
            "many objects",
            "photos",
            "surreal",
            "people",
            "human",
            "face",
            ],
            "guidance_scale": 20,
            "temperature":0.5
              })

    image_format = imghdr.what(None, h=image_bytes)

    if image_format is None:
        raise ValueError("Unsupported image format or corrupted image data")

    image = Image.open(io.BytesIO(image_bytes))

    return image
```

Figure 5.42: generate logo function

Finally, the collected data, including the company description, brand name, and generated logo images, are forwarded to the next step in the branding process.

```
return render_template('brand-section4.html',
businessDescription=company_description, brandName=brand_name,
color1=color1, color2=color2, color3=color3,  generated_logo=generated_logo)
```

Figure 0.43: Sending collected data to the Rendering Page

Brand Identity Section 4:
The HTML code represents the final step in the process of creating a brand identity on the MOSAWEEQ platform. In this step, the user chooses the company logo from a set of images generated by the AI model. The selected logo's value is stored in a hidden input field named `logo_image`. The form includes a submit button labeled "Next" that directs the user to the next step when clicked, finalizing the branding process. As shown in Figure (Brand identity section4 HTML Form)

```
<div class="row">
    <div class="main-content">
        <div class="w3-container w3-teal">
            <h3 style="direction: rtl ;color: white ;padding-bottom: 10px;">انشئ هويتك التجارية</h3>
        </div>

        <div id="section4" class="content-section active">
            <form class="rectanglee" method="post" action="/branding/step4" id="logoForm">
                <div class="text">
                    <p>اختر شعار شركتك:</p>
                </div>
                <div class="image-grid">
                </div>
                <input type="hidden" name="logo_image" id="logo_image" value="">
                <!-- Next Button -->
                <div class="w3-container w3-teal MoveButton">
                    <input type="submit" type="button" class="next" value="التالي"></input></a>
                </div>
            </form>
        </div>
    </div>
```

Figure 0.44: Brand identity section4 HTML Form

The code defines the `branding_step4` route for handling the final step in the brand identity creation process on the MOSAWEEQ platform. When the user submits the form via POST, it retrieves the selected logo image and user ID from the session. The code then collects all the previously gathered branding information (business category, description, brand name, and colors) from the session.

The code connects to a SQLite database and inserts this information into the appropriate tables: `Business`, `Category`, `BrandIdentity`, and `ColorPalette`. If the data is successfully saved to the database, the user is directed to the next step by rendering the `brand-section5.html` template, which displays the selected logo and other brand details. If an error occurs during database operations, the transaction is rolled back, and an error message is displayed.

Figure 0.45: branding step4 route

Brand Identity Section 5:
The HTML code serves to present a comprehensive summary of the user's branding decisions made throughout the MOSAWEEQ platform's brand identity creation process. This summary encapsulates crucial elements such as the chosen logo, brand name, brand colors, and a brief description of the business.
To populate this summary, the code retrieves information about the brand identity choices from the database. It queries relevant tables to fetch data such as the selected logo, brand name, brand colors, and business description.

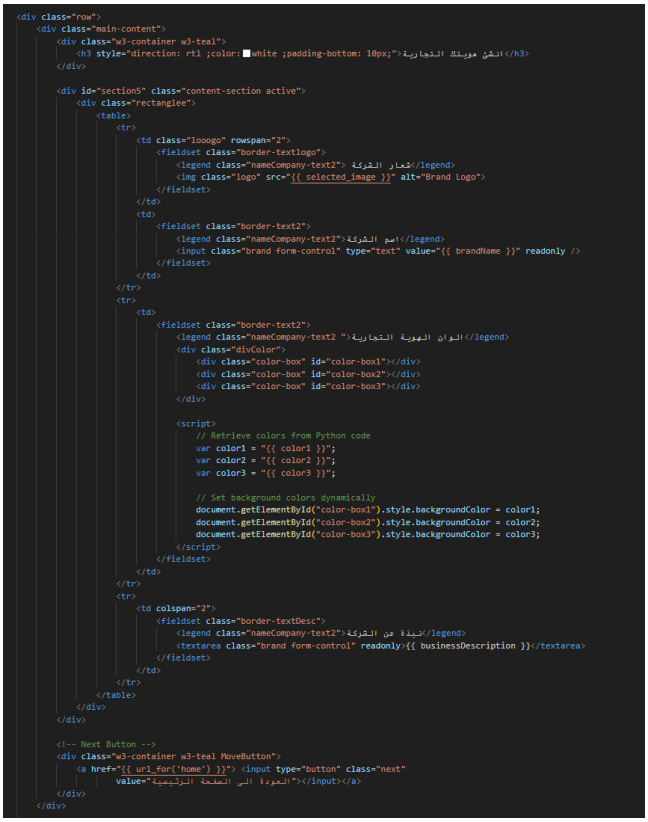


Figure 0.46: Brand identity represented in HTML

5.2.3.8 Calendar Page:

For the front end of the calendar page, the HTML code enhances MOSAWEEQ platform by integrating Google Calendar functionality, providing users with a comprehensive view of upcoming events and their generated description, facilitating content generation based on these events. as shown in Figure (Calendar Page HTML)

Figure 0.47: Calendar Page HTML

Integration with Google Calendar API:

We have secured access from Google to use their API to display our calendar, "MOSAWEEQ", on our website by obtaining a credentials ID. This integration allows us to fetch and display calendar events seamlessly. Below is an implementation snippet that demonstrates how we utilize Google's Calendar API:

```python
# Route to render the calendar content page
from google.oauth2 import service_account
from googleapiclient.discovery import build
from datetime import datetime, timedelta, timezone
import time

@app.route("/generateContentWithEvent", endpoint="generateContentWithEvent")
def generateContentWithEvent():
    try:
        # Load service account credentials from JSON file
        SERVICE_ACCOUNT_FILE = 'mosaweeq-154a335ba501.json'
        SCOPES = ['https://www.googleapis.com/auth/calendar.readonly']
        credentials = service_account.Credentials.from_service_account_file(
            SERVICE_ACCOUNT_FILE, scopes=SCOPES)

        # Create a service object for making API requests
        service = build('calendar', 'v3', credentials=credentials)
```

Figure 0.48: Utilizing Google's Calendar API

To read the events, we downloaded a JSON file for credentials to access the upcoming events. We then modified the code to fetch the next three events occurring within the next two months (60 days). After reading these upcoming events, we send their names to a function called generate_arabic_description, which uses Google Generative AI to create a short description for each event to display to the user, and then stores the description in the database with the name and the start date of the event.

```python
    # Define the time range for events (one month from now)
    now = datetime.now(timezone.utc)
    end_time = now + timedelta(days=60)
    time_min = now.isoformat()
    time_max = end_time.isoformat()

    # Initialize a list to store upcoming events information'''
    calendar_id = '192a947afbf35a7f834ecbb9832bbadadcb981bfe69721368e8ab6d39daf99b8@group.calendar.google.com'
    all_events = []
    batch = service.new_batch_http_request()
    batch.add(service.events().list(
        calendarId=calendar_id,
        timeMin=time_min,
        timeMax=time_max,
        maxResults=10,  # Adjust the maxResults as needed
        singleEvents=True,
        orderBy='startTime'
    ), callback=lambda request_id, response, exception: all_events.extend(response.get('items', [])))
    batch.execute()
    all_events.sort(key=lambda x: x['start'].get('dateTime', x['start'].get('date')))
```

Figure 0.49: Code fetches the next three events occurring within 60 days using a downloaded JSON credentials file.

```python
    # Translate event names to Arabic and generate descriptions
    for event in all_events[:3]:  # Only process the first 3 events to reduce API usage
        event_name = event['summary']
        start_date = event['start'].get('dateTime', event['start'].get('date'))

        # Check if the event exists in the database
        cursor.execute("SELECT * FROM Event WHERE Event_name = ?", (event_name,))
        existing_event = cursor.fetchone()

        if existing_event:
            # If the event exists, retrieve its description from the database
            arabic_description = existing_event[2]  # Assuming description is the third column
        else:
            # If the event does not exist, generate the description and insert into the database
            arabic_description = generate_arabic_description(event_name)
            cursor.execute("INSERT INTO Event (Event_name, Event_description, Start_date) VALUES (?, ?, ?)",
                           (event_name, arabic_description, start_date))
            conn.commit()
            time.sleep(1)

        # Add event information to upcoming_events_info
        upcoming_events_info.append({'start': start_date, 'summary': event_name, 'description': arabic_description})
        #upcoming_events_info.append({'start': start_date, 'summary': arabic_translation})

    # Pass the upcoming events information to the template
    return render_template("calendar.html", upcoming_events=upcoming_events_info)
```

Figure 0.50: Code Snippet shows the generation of the Arabic description of the three upcoming event, then storing them in the database.

```python
def generate_arabic_description(event_name):
    input_text = f"اكتب لي شرحًا قصيرًا لمناسبة {event_name}، 24 هو أقصى عدد من الكلمات."
    convo = model.start_chat(history=[])
    convo.send_message(input_text)
    output = convo.last.text.split('\n')
    arabic_description = output[0].split("is")[-1].strip()
    return arabic_description
```

Figure 0.51: generate_arabic_description function, to generate an arabic description of the event.

5.2.3.9 Customize content Page:

The HTML code represents a form used for creating new content on the platform. Upon loading the page, a JavaScript function named `populateDropdown()` is invoked. This function sends an AJAX request to the backend to retrieve a list of brand names for the user. Once the data is received, the dropdown menu for selecting a brand is dynamically populated with the fetched brand names. We use this to retrieve the data for the brand selected then use the data for generating marketing taglines and generating visual advertisement design.

The form allows users to select a brand from the dropdown menu and enter a description for the advertising content they wish to create. Users input their advertisement description into a textarea element. Additionally, the form includes two buttons, each with an `onclick` attribute that dynamically sets the form action based on the button clicked.

The first button, labeled "انشئ تصميم اعلان", sets the form action to "/createPosts", indicating that the user wants to create a visual advertisement design. The second button, labeled "انشئ عبارات تسويقية", sets the form action to "/createTaglines", indicating that the user wants to generate marketing taglines.

```html
<div class="CustomizeContainer">
    <h1>انشاء محتوى جديد</h1>
    <div class="sq">
        <script>
            // Function to fetch data from the backend and populate the dropdown
            function populateDropdown() {
                // Make AJAX request to get data from the backend
                var xhr = new XMLHttpRequest();
                xhr.open("GET", "/get_brands", true);
                xhr.onreadystatechange = function () {
                    if (xhr.readyState === 4 && xhr.status === 200) {
                        var brands = JSON.parse(xhr.responseText);
                        var dropdown = document.getElementById("brand");
                        // Clear existing options
                        dropdown.innerHTML = "";
                        // Add options to the dropdown
                        brands.forEach(function (brandName) {
                            var option = document.createElement("option");
                            option.value = brandName;
                            option.textContent = brandName;
                            dropdown.appendChild(option);
                        });
                    }
                };
                xhr.send();
            }
            // Call the populateDropdown function when the page loads
            window.onload = populateDropdown;
        </script>
        <form method="POST" id="shared_form">
            <div class="brannnd">
                <p id="ChooseAdParagraph" for="brand">اختر العلامة التجارية:</p>
                <select name="brand" id="brand">
                    <!-- Dropdown options will be added dynamically here -->
                </select>
            </div>
            <p id="DescriptionParagraph1">:ادخل وصف المحتوى الإعلاني</p>
            <textarea id="input_text" name="input_text" placeholder="مثال.. اريد عمل إعلان عن خواتم الفضة تحديداً "
                maxlength="200" required>{{ input_text }}</textarea><br>
            <div class="button-container">
                <button type="submit" onclick="setAction('/createPosts')" class="button">انشئ تصميم
                    اعلان!</button>
                <button type="submit" onclick="setAction('/createTaglines')" class="button">انشئ عبارات
                    تسويقية</button>
            </div>
        </form>
        <script>
            function setAction(action) {
                document.getElementById('shared_form').setAttribute('action', action);
            }
        </script>
    </div>
</div>
```

Figure 0.52: HTML form enables users to select a brand and input advertising descriptions, with dynamic options to create visual ads or generate marketing taglines

This Flask route, accessible at `/createContent`, is responsible for handling the submission of the content creation form. Upon receiving a POST request, it checks if the user is logged in by verifying the presence of their user ID in the session. If the user is authenticated, the function retrieves the input text (the description of the content) and the brand name from the form submission.

Next, it establishes a connection to the database and retrieves the business ID associated with the selected brand name. It then queries the database to fetch the business description and category name corresponding to the retrieved business ID.

This route serves as the backend logic for generating content based on the user's input text and the associated brand information retrieved from the database.

```python
# Route to re generating form submission
@app.route('/createContent', methods=['POST'])
def createContent():
    if request.method == 'POST':
        if 'user_id' in session:
            user_id = session['user_id']
            # Retrieve the tagline from the form submission
            input_text = request.form['input_text']
            brand_name=session['brand_name']

            # Connect to the database
            con = sqlite3.connect('Mosaweeq.db')
            cursor = con.cursor()

            cursor.execute("""
               SELECT b.Business_ID
               FROM Business b
               INNER JOIN BrandIdentity bi ON b.Business_ID = bi.Business_ID
               WHERE bi.Brand_name = ?
               """, (brand_name,))
            business_id_result = cursor.fetchone()

            # Check if a result is found
            if business_id_result:
                # Extract the Business_ID value
                business_id = business_id_result[0]

            print("this is the business ID", business_id)

            cursor.execute("""
               SELECT b.Business_description, c.Category_Name
               FROM Business b
               INNER JOIN Category c ON b.Business_ID = c.Business_ID
               WHERE b.Business_ID = ?
               """, (business_id,))

            # Fetch the result
            business_info = cursor.fetchone()
```

Figure 0.53: Flask Route for Content Creation Form Submission and Brand Information Retrieval

Create Taglines

To generate marketing taglines, we employed prompt engineering, a technique that involves designing inputs for AI tools to produce optimal outputs. Our prompts were structured to request marketing content in two scenarios: when there is a selected event and when the user requests general marketing content not tied to a specific event. The prompt sent to the AI model is chosen after checking if there was an event name stored in the session. Additionally, we incorporated emojis and retrieved brand data from the database to ensure the content's relevance to the brand. [reference]
Below are the prompts used:

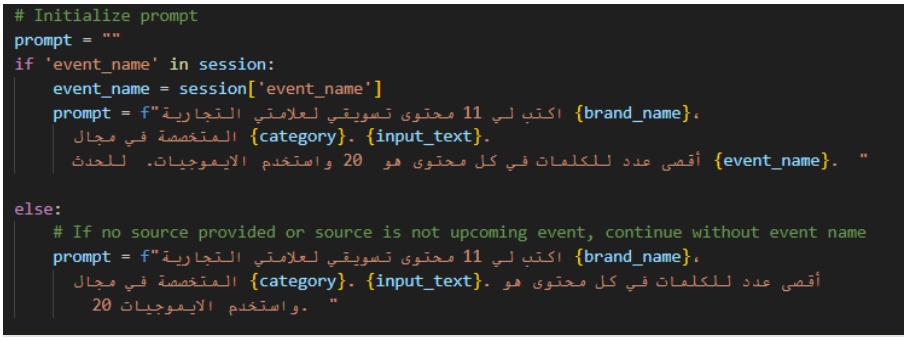


Figure 0.54: Tagline prompt

The following code shows the communication with the AI model to generate marketing taglines:

```
convo = model.start_chat(history=[])
convo.send_message(prompt)
output = convo.last.text.split('\n')
taglines = []
```

Figure 0.55: Code for interacting with the AI model

As we did in generating the brand names, we initiated a new chat session with the AI model,then we sent the designed prompt to the AI model. Finally, we retrieved the model's response using output = convo.last.text.split('\n'), which splits the response into a list of lines for easier processing.
After obtaining the results from the AI model, we further manipulate the generated text to enhance its visual appeal for the user. This involves removing unnecessary white spaces, headlines, dots, and asterisks, ensuring the content is clean and engaging.

```
if len(output) >= 2:
    for line in output[1:]:
        parts = line.split('. ')
        if len(parts) > 1:
            tagline = parts[1].strip()
            tagline = tagline.replace('**', '')
            taglines.append(tagline)
```

Figure 0.56: Enhancing AI-generated text

Create Taglines Page:

The HTML form enables users to select marketing taglines for their brand. It provides a list of taglines generated by the AI model, allowing users to choose from them. Users can also input their content description. Upon submission, the form generates marketing taglines based on the provided description. Additionally, there's a popup container that displays the full taglines with options to cancel or save the selected tagline.

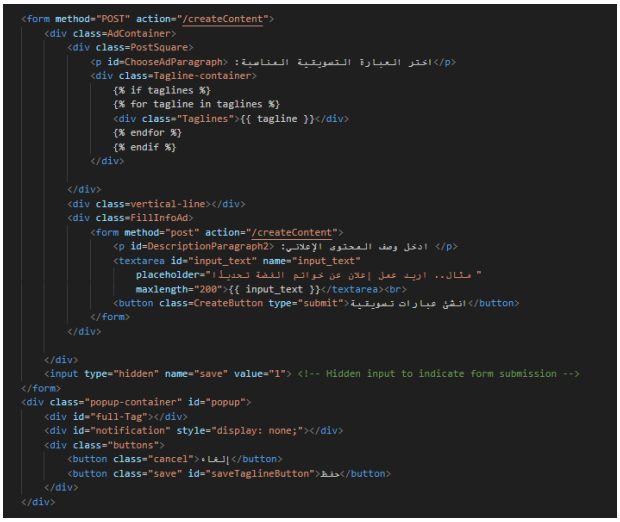


Figure 0.57: Displaying list of generated marketing taglines for user to choose preferred one.

The `/save_tagline` route is responsible for saving the selected tagline in the database. Upon receiving a POST request, it retrieves the tagline from the JSON data sent in the request. The route then establishes a connection to the database and retrieves the business ID associated with the current session's brand name. If an event name is stored in the session, it also retrieves the event ID based on the event name.

Subsequently, it inserts the tagline into the `Content` table of the database. If an event name is present in the session, the tagline is associated with both the business ID and the event ID. Otherwise, it's only associated with the business ID. Finally, the route returns a JSON response indicating that the tagline has been successfully saved.

```python
@app.route("/save_tagline", methods=["POST"])
def save_tagline():
    tagline = request.json.get("tagline")
    # Connect to the database
    con = sqlite3.connect('Mosaweeq.db')
    cursor = con.cursor()

    brand_name=session['brand_name']
    cursor.execute("""
            SELECT Business_ID
            FROM BrandIdentity
            WHERE Brand_name = ?
        """, (brand_name,))
    business_id_result = cursor.fetchone()

    if business_id_result:
        business_id = business_id_result[0]
    else:
        print("No business found for the given brand name")

    if 'event_name' in session:
        # Retrieve the Event_ID based on the event name from the session
        cursor.execute("""
            SELECT Event_ID
            FROM Event
            WHERE Event_name = ?
        """, (session['event_name'],))
        event_id_result = cursor.fetchone()

    # Save tagline in the database
    if 'event_name' in session:
        event_id = event_id_result[0]
        cursor.execute("""
        INSERT INTO Content (Text_content, Business_ID, Event_ID)
        VALUES (?, ?, ?)
        """, (tagline, business_id, event_id))
        con.commit()
    else:
        cursor.execute("""
            INSERT INTO Content (Text_content, Business_ID)
            VALUES (?, ?)
        """, (tagline, business_id))
        con.commit()

    return jsonify({"message": "Tagline saved successfully"}), 200
```

Figure 0.58: Process flow of saving taglines in the database.

Create Posts

When a user interacts with the system to create a post, they have two options: uploading a product image or providing only a description. Let's start with the scenario where the user uploads a product image.
In this case, the system begins by generating a prompt for creating the post, specifically focusing on appreciating the background of the uploaded image. As shown below:

```
description = "Translate this to english 9 times\
(An background, no object, soft lighting, design, showcasing \
intricate details and superb clarity. simple, high-quality \
photography). Each translation start with ^^^^ "

advertisement = generate_advertisement(description)
```

Figure 0.59: background image prompt

This prompt is translated into nine different sentences using text generation models to ensure variety and uniqueness in the resulting posts.

```
temp_dir = tempfile.gettempdir()
os.makedirs(temp_dir, exist_ok=True)  # Ensure the temporary directory exists
image_filename = secure_filename(image_file.filename)
image_path = os.path.join(temp_dir, image_filename)
image_file.save(image_path)
```

Figure 0.60: Image processing code securely saves uploaded images for further processing

Once the prompt is generated, the system proceeds to process the uploaded image. The code block responsible for this task utilizes the system's temporary directory to securely save the image. It generates a secure filename for the image to mitigate potential security risks and then saves the image to the temporary directory for further processing.

```
if advertisement is not None:  # Check if advertisement is not None
    clean = clean_translations(advertisement)
    for idx in range(1,len(clean)):
        adver = clean[idx]
        image = process_images(image_path,adver,logo_url)
        if image is not None:
            buffer = io.BytesIO()
            image.save(buffer, format="JPEG")
            base64_encoded_image = base64.b64encode(buffer.getvalue())
            generated_image = base64_encoded_image.decode('utf-8')
            generated_images.append(generated_image)

    return jsonify({'generated_images': generated_images})
```

Figure 0.61: Image processing creates post images from prompts and uploads, encoded for storage

After the image is processed and securely saved, the system utilizes the prompt and cleaning the prompt to generate the post. This process generates post images based on the prompt and uploaded image and encoding these images into base64 format for storage and compatibility purposes.

```python
def process_images(image_path,description,logo_url):
        try:
            # Read the image file as bytes
            with open(image_path, 'rb') as f:
                image_data = f.read()
            # Encode the image data to base64
            base64_image = base64.b64encode(image_data).decode("utf-8")
            # Load the overlay image using pipeline
            pipe = pipeline("image-segmentation", model="briaai/RMBG-1.4", trust_remote_code=True)

            overlay_image = pipe(base64_image)  # applies mask on input and returns a pillow image
```

Figure 0.62: removing the background and isolating the object

The `process_images` function is responsible for the condition where the user has uploaded an image. Initially, it reads the image file from the provided path and encodes the image data to base64 format. This encoded image is then processed using an image segmentation model from the "briaai/RMBG-1.4" pipeline to apply a mask on the input image, effectively removing the background and isolating the object.

```python
    image_bytes = query({
        "inputs": description,
        "negative_prompt": [···
        "temperature":1,
        })
    # access the image with PIL.Image for example
    imagebackground = Image.open(io.BytesIO(image_bytes))
    # marge both image
    scaling_factor = min(imagebackground.width / overlay_image.width, imagebackground.height / overlay_image.height)
    # Resize overlay image
    overlay_image_resized = overlay_image.resize((int(overlay_image.width * scaling_factor),
                                                  int(overlay_image.height * scaling_factor)), Image.BILINEAR)
```

Figure 0.63: image generation process

Next, the function generates a background image based on the provided description by querying an image generation model. The overlay image and the generated background image are merged, ensuring that the overlay image is resized to fit appropriately onto the background. The resizing is done proportionally to maintain the aspect ratio, and the overlay image is placed on the right side of the background image.

```python
# Create a new image to store the result
result_image = imagebackground.copy()

# Paste the resized overlay image onto the background image at the right
x = imagebackground.width - overlay_image_resized.width
y = (imagebackground.height - overlay_image_resized.height) // 2
result_image.paste(overlay_image_resized, (x, y), overlay_image_resized)

# Open the logo image and convert to RGBA (adding an alpha channel)
logo_img = Image.open(io.BytesIO(logo_url)).convert("RGBA")
logo_h, logo_w = logo_img.size

# Create a mask with a circular shape
mask = Image.new('L', (logo_w, logo_h), 0)
draw = ImageDraw.Draw(mask)
draw.pieslice([0, 0, logo_w, logo_h], 0, 360, fill=255)

# Apply the mask to the logo image
logo_img.putalpha(mask)

# Resize the logo image to the desired size
desired_logo_size = (100, 100)
resized_logo_img = logo_img.resize(desired_logo_size)

# Paste the resized logo onto the result image
result_image.paste(resized_logo_img, (20, 5), resized_logo_img)
```

Figure 0.64: Overlay image merged onto a generated background

After merging the overlay and background images, the function processes the logo image associated with the brand. It opens the logo image sent as a parameter, converts it to RGBA format to add an alpha channel, and creates a circular mask for the logo. This mask is applied to the logo image, which is then resized to the desired dimensions. The resized logo is pasted onto the merged image, resulting in the final composite image.

Now, let's consider the scenario where the user doesn't upload a product image, providing only a description for the post.
In this scenario, we use the same data retrieval from the database. This includes the brand name, logo URL, and category name. This information is utilized in designing the prompt based on the description provided by the user. As shown below:

```
description = ("Translate this to English 9 times, fill hashtags (An image of # with soft lighting "
    "and an exclusive # design for " + category_name + ", showcasing intricate details "
    "and superb clarity. The background, whether simple or ornate, enhances the product's "
    "beauty through high-quality photography). Each translation starts with ^^^^ " + input_text)
```

Figure 0.65: post prompt

After designing the prompt, the process remains similar to the scenario where a product image is uploaded. The descriptions are cleaned and sent to the `generate_image` function.

```
    # Modify the description slightly for each iteration to make it unique
    translated_description = generate_advertisement(description)
    if translated_description:
        translations = clean_translations(translated_description)

    else:
        logging.error("translation adv is None.")

    # Loop through the translations list
    for idx in range(1,len(translations)):
        # Get the translation at the current index
        translation = translations[idx]
        # Generate image based on the translated description
        generated_image = generate_image(translation,logo_url)

        if generated_image is not None:
            # Encode the image to base64
            buffer = io.BytesIO()
            generated_image.save(buffer, format="JPEG")
            base64_encoded_image = base64.b64encode(buffer.getvalue())
            generated_image = base64_encoded_image.decode('utf-8')
            generated_images.append(generated_image)
        else:
            logging.error("Generated image is None.")

    # Return JSON response with generated images
    return jsonify({'generated_images': generated_images})
```

Figure 0.66: Image processing creates post images from prompts, encoded for storage

The `generate_image` function is responsible for generating marketing posts based on user-provided descriptions. Its behavior adapts based on whether there's a selected event. If an event is chosen, the function customizes the prompt by integrating the user's description with the event name retrieved from the session data. This customized prompt ensures that the text to be placed within the image later aligns with the selected event, providing contextual relevance.

```
if 'event_name' in session:
    event_name = session['event_name']
    txt = " اكتب اعلان تسويقي من هذا الوصف بدون اي علامات "+ prompt+" المناسبة لهذا كلمات. 5 للكلمات عدد اقصى" + event_name
else:
    txt = " اكتب اعلان تسويقي من هذا الوصف بدون اي علامات "+ prompt+" المناسبة لهذا كلمات. 5 للكلمات عدد اقصى"

text = generate_advertisement(txt)
```

Figure 0.67: check if the user selected an event or not

Next, the function initiates the image generation process using the generated prompt as input. It also repeats the process of adding the logo to the generated image. However, it specifically utilizes the customized prompt to generate the text that will be incorporated into the image. Both the image and the generated text are then passed to the function "draw_text_with_rectangle" to further enhance their visual appeal.

```
def draw_text_with_rectangle(image, text):
        reshaped_text = arabic_reshaper.reshape(text)
        bidi_text = get_display(reshaped_text)
        font = ImageFont.truetype('fonts/18KhebratMusamimRegular.ttf', size=50)
        d = ImageDraw.Draw(image)

        # Get the bounding box of the text
        text_bbox = d.textbbox((0, 0), bidi_text, font=font)

        # Calculate the dimensions of the text
        text_width = text_bbox[2] - text_bbox[0]
        text_height = text_bbox[3] - text_bbox[1]

        # Define the coordinates for the rectangle
        rect_x = image.width - text_width - 40  # Adjust for padding
        rect_y = image.height - text_height - 60  # Adjust for padding
        rect_width = text_width + 20  # Add padding to both sides of the rectangle
        rect_height = text_height + 20  # Add padding to both sides of the rectangle

        # Draw the rectangle behind the text
        d.rectangle([(rect_x, rect_y), (rect_x + rect_width, rect_y + rect_height)], fill='white')

        # Draw the text
        text_x = rect_x + 10  # Adjust as needed to provide padding inside the rectangle
        text_y = rect_y + 5  # Adjust as needed to provide padding inside the rectangle
        d.multiline_text((text_x, text_y), bidi_text, font=font, fill='green', spacing=151, align='right')
        return image
```

Figure 0.68: draw_text_with_rectangle function

The function takes steps to enhance the visibility of the tagline text by creating a text box behind it. This ensures that the tagline stands out prominently within the marketing post. The tagline text is meticulously drawn onto the image using Arabic text formatting techniques.

Create Post Page:

The HTML form enables users to create advertisements by inputting a description or uploading an image file. Upon submission, the form data, including the description and image, is sent to the server for processing. Additionally, there's a hidden popup container for displaying full-size images when clicked, with options to cancel or save the image.

```
<form method="post" id="adForm" enctype="multipart/form-data" action="/">
    <div class="AdContainer">
        <div class="PostSquare">
            <div class="image-container" id="resultContainer">
                {% if generated_images %}
                {% for image in generated_images %}
                <img src="{{ image }}" alt="generated image" class="image-box">
                {% endfor %}
                {% endif %}
            </div>
        </div>
        <div class="FillInfoAd">
            <!-- Description input -->
            <p id="DescriptionParagraph2"> :ادخل وصف المحتوى الإعلاني </p>
            <textarea type="text" id="description2" name="input_text"
                placeholder="مثال.. اريد عمل إعلان عن خواتم الفضة تجديداً! " maxlength="200"
                required>{{ input_text }}</textarea>
            <div class="yes">
                <!-- File input button -->
                <label for="imag" class="btn_upload">اختر صورة
                    <input type="file" id="imag" name="imag" title="" class="input-img" />
                </label>
            </div>
            <!-- Image preview container -->
            <div id="image-preview-container">
                <img id="ImgPreview" src="" class="preview1" />
                <!-- Remove image button -->
                <input type="button" id="removeImage1" value="x" class="btn-rmv1" />
            </div>
            <!-- Button to submit form -->
            <button type="submit" class="CreateButton">انشئ اعلان</button>
        </div>
    </div>
</form>

<!-- Popup container for displaying full image -->
<div class="popup-container" id="popup">
    <img src="" alt="Full Image" id="full-image">
    <div id="notification" style="display: none;"></div>
    <div class="buttons">
        <button class="cancel">إلغاء</button>
        <button class="save">حفظ</button>
    </div>
</div>
```

Figure 0.69: create post HTML form

The `download_image()` route in the Flask application handles the downloading of images. When a POST request is made to this route, it retrieves the image URL from the form data. It then extracts necessary data from the session, such as the event name and user ID, to associate the image with the correct business and event in the database.

The route connects to the database, retrieves the business ID associated with the user, and optionally the event ID if available. It then inserts the image URL along with the business and event IDs into the Content table.

Finally, the route sends the temporary image file for download to the client as an attachment. This functionality allows users to download the generated images for further use.

```
@app.route('/download', methods=['POST'])
def download_image():
    image_url = request.form.get('image_url')
    # Convert base64 image to bytes
    event_name = session.get('event_name')
    user_id = session.get('user_id')

    # Connect to the database
    con = sqlite3.connect('Mosaweeq.db')
    cursor = con.cursor()

    if user_id:
        cursor.execute("""SELECT Business_ID FROM Business WHERE User_ID = ? """, (user_id,))
        business_id_result = cursor.fetchone()

        if business_id_result:
            business_id = business_id_result[0]
        else:
            print("Business ID not found for the user")
            return

        if event_name:
            cursor.execute("""SELECT Event_ID FROM Event WHERE Event_name = ? """, (event_name,))
            event_id_result = cursor.fetchone()

            if event_id_result:
                event_id = event_id_result[0]
            else:
                print("Event not found")
                event_id = None
        else:
            event_id = None

        # Execute the INSERT statement to add the image to the Content table
        cursor.execute("""
            INSERT INTO Content (image_url, business_id, Event_ID)
            VALUES (?, ?, ?)
            """, (image_url, business_id, event_id))

        # Commit the transaction to save changes to the database
        con.commit()
    else:
        print("User ID not found in session")

    image_bytes = base64.b64decode(image_url.split(',')[1])
    # Create a temporary file to save the image
    with tempfile.NamedTemporaryFile(delete=False) as temp_image:
        temp_image.write(image_bytes)
        temp_image_path = temp_image.name

    # Send the file for download
    return send_file(temp_image_path, as_attachment=True)
```

Figure 0.70: Process flow of saving posts in the database.

5.3 Conclusion

In this chapter we explained all of the essential functionalities required to make our website run successfully from integrating AI models, connecting the database until the construction of the website. In the next chapter, we will test the system to evaluate and verify it.

Chapter 6: Testing

6.1 Introduction

The testing phase of software development constitutes a pivotal stage in the software lifecycle, serving as a meticulous evaluation process aimed at ensuring the quality, reliability, and functionality of the developed software. This chapter will elucidate five distinct testing types of integral to software development: unit testing, integration testing, compatibility testing, usability testing, and system testing.

6.2 Unit Test

Unit testing is an essential aspect of software development. It involves checking individual parts of the software to ensure they function correctly on their own. These parts, such as functions or classes, are tested independently to verify their accuracy and reliability. In this section, unit testing is employed to evaluate the functionality of the email existence verification feature on the signup page. This ensures that users cannot register with an email address that is already associated with an existing account. Additionally, unit testing is conducted on the password validation mechanism to enforce adherence to specified requirements. This ensures that users cannot register with a password that fails to meet the predefined criteria.

6.2.1 Testing the Validation of Password Method

This is the first unit test created for the signup page. It focuses on validating passwords according to specific requirements. The goal is to ensure that users cannot sign up with weak or insecure passwords.

Here's what the test does:

Test Valid Passwords:

It checks if passwords meeting the requirements are accepted.

Examples include "Password1!" and "Strong123@".

These passwords have a minimum length of 8 characters, contain at least one digit, and one special character.

Test Invalid Passwords:

It verifies that passwords not meeting the requirements are rejected.

Examples include "pass" (too short), "password" (lacks special characters or digits), "WeakPass" (no special character), "Weak1234" (no special character), "Weak@Pass" (no digit), "12345678" (no letter), and "!@#$$%^&*" (no letter or digit).

```python
import unittest
from app import validate_password

class TestPasswordValidation(unittest.TestCase):

    def test_valid_password(self):
        # Test valid passwords
        self.assertTrue(validate_password("Password1!"))  # Minimum length with digit and special character
        self.assertTrue(validate_password("Strong123@"))  # Longer length with digit and special character

    def test_invalid_password(self):
        # Test invalid passwords
        self.assertFalse(validate_password("pass"))  # Too short
        self.assertFalse(validate_password("password"))  # No special character or digit
        self.assertFalse(validate_password("WeakPass"))  # No special character
        self.assertFalse(validate_password("Weak1234"))  # No special character
        self.assertFalse(validate_password("Weak@Pass"))  # No digit
        self.assertFalse(validate_password("12345678"))  # No letter
        self.assertFalse(validate_password("!@#$$%^&*"))  # No letter or digit

if __name__ == '__main__':
    unittest.main()

```

Figure 6.1: testing password validation

Figure 6.2: Result

6.2.2 Testing the Existence of Email Method

This unit test is designed to verify whether the signup page correctly checks if an email already exists in the database. The purpose is to prevent users from registering with an email that's already associated with an existing account.

Here's what the test does:

Test Existing Email:

It checks if the function correctly identifies an existing email in the database.

An example email, 'rr.ranashi14@gmail.com', is used as a known existing email in the test database.

The test expects the function to return True, indicating that the email already exists.

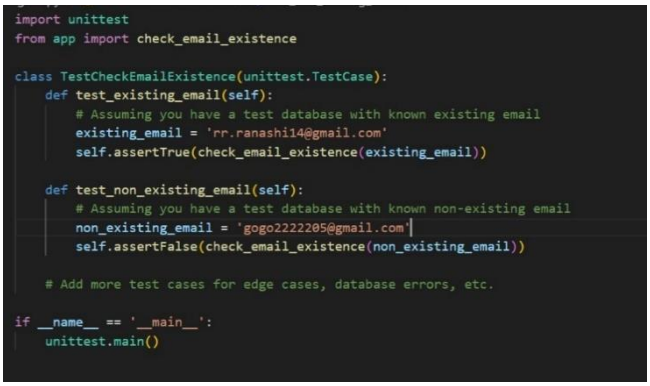
Test Non-Existing Email:

It verifies if the function correctly identifies a non-existing email in the database.

Another example email, 'gogo2222205@gmail.com', is used as a known non-existing email in the test database.

The test expects the function to return False, indicating that the email doesn't exist and can be used for registration.

These tests ensure that the email existence check function operates as intended, allowing the signup page to prevent duplicate email registrations effectively.



```python
import unittest
from app import check_email_existence

class TestCheckEmailExistence(unittest.TestCase):
    def test_existing_email(self):
        # Assuming you have a test database with known existing email
        existing_email = 'rr.ranashi14@gmail.com'
        self.assertTrue(check_email_existence(existing_email))

    def test_non_existing_email(self):
        # Assuming you have a test database with known non-existing email
        non_existing_email = 'gogo2222205@gmail.com'
        self.assertFalse(check_email_existence(non_existing_email))

    # Add more test cases for edge cases, database errors, etc.

if __name__ == '__main__':
    unittest.main()
```

Figure 0.3: Testing email existence

Figure 6.4: Result

6.3 Integration Test

Integration testing stands as a pivotal phase in software development, where the interaction between various components is scrutinized for seamless operation. Unlike unit testing, which assesses individual elements in isolation, integration testing focuses on the combined functionality of integrated modules.

6.3.1 Integration Test for Correct Email and Password

To verify the successful login functionality using an existing email and password in the MOSAWEEQ database, execute the following `curl` command. This command simulates a login attempt with valid credentials. When executed, it results in the successful rendering of the homepage, indicating that the login process is working correctly.

```
C:\Users\rnosh>curl -X POST -d "email=lama.albelewi@outlook.com&password=0909" http://127.0.0.1:5000/login
```

Figure 6.5: Curl command for testing with correct email and password

```
<!DOCTYPE html>
<html lang="en" xmlns="http://www.w3.org/1999/xhtml">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title> [illegible] </title>
    <link rel="icon" type="images/MosaweeqIcon.png" href="/static/images/MosaweeqIcon.png">
    <link rel="stylesheet" href="/static/css/styles3.css" />
    <link rel="stylesheet" href="/static/css/Home.css" />
    <link rel="preconnect" href="https://fonts.googleapis.com" />
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link href="https://fonts.googleapis.com/css2?family=Cairo" rel="stylesheet">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css" />
    <link href="https://fonts.googleapis.com/css2?family=Cairo:wght@400;700&display=swap" rel="stylesheet" />
</head>

<body>
    <header class="header">
        <div class="container">
            <div class="left-section">

                <!-- Display user information if user is logged in -->
                <a href="/logout" onclick="return confirmLogout()"><button id="signOut">[illegible]
                        [illegible]</button></a>
                        <a href="/profile" class="profile-info" id="profile">
                            <img src="/static/images/profileIcon.jpg" class="userIconn" alt="Profile Icon">
                            <span class="username">Lama Albelewi</span>
```

Figure 0.6: Result: Rendering the home page

6.3.2 Integration Test for Incorrect Email or Password

To test the login functionality with an existing email but an incorrect password, use the following `curl` command. This command simulates a login attempt with an invalid password. When executed, it results in the rendering of the login page, indicating an unsuccessful login attempt and confirming that the system correctly handles incorrect credentials.

```
C:\Users\rnosh>curl -X POST -d "email=lama.albelewi@outlook.com&password=000" http://127.0.0.1:5000/login
```

Figure 0.7: Curl command for testing with incorrect password

```
C:\Users\rnosh>curl -X POST -d "email=lama.albelewi@outlook.com&password=000" http://127.0.0.1:5000/login
 <!DOCTYPE html>
<html lang="ar">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link rel="stylesheet" href="/static/css/signin.css" />
    <link href="https://fonts.googleapis.com/css2?family=Cairo" rel="stylesheet">
    <title>[illegible]</title>
    <link rel="icon" type="images/MosaweeqIcon.png"
        href="/static/images/MosaweeqIcon.png" />
    <style>
        .error-message {
            color: red;
            margin-top: 10px;
            direction: rtl;
        }
    </style>
</head>

<body>

    <div class="container">
        <div class="row">
            <div class="col left-section">
```

Figure 0.8: Result: Rendering the sign in page

6.4 Compatibility Test:

Compatibility testing ensures that software performs seamlessly across diverse platforms, devices, and environments. It verifies that the software functions consistently and effectively, regardless of the user's setup.
A compatibility test was conducted on Windows devices using both Chrome and Edge browsers, and Mosaweeq passed the test successfully.

Device	Browser	Pass/Fail
Windows	Google Chrome Version 124.0.6367.203 (Official Build) (64-bit)	Pass
	Microsoft Edge Version 124.0.2478.80 (Official build) (64-bit)	Pass

Table 0.1: Compatibility test

6.5 System Test:

The summary of system test cases, encompassing both front-end and back-end components, along with their corresponding expected and actual outputs, is outlined in Table 6.2. The results depicted in the table affirm that all test cases have been successfully passed, demonstrating that MOSAWEEQ system is operating in accordance with expectations.

Test Case ID	Description	Test Data	Expected Output	Actual Output	Pass / Fail
Test Scenario 1: Check the Sign-up Functionality					
1.1	Sign up with valid email	Email	Signup successfully and redirect to the homepage	Signup successfully and redirect to the homepage	pass
1.2	Sign up with wrong format email	Email	Error message displayed	Error message displayed	
1.3	Sign up with existing email	Email	Error message displayed	Error message displayed	pass
1.4	Signup with password contains more than 7 characters and at least 1 special character and digit	password	Signup successfully and redirect to the homepage	Signup successfully and redirect to the homepage	pass
1.5	Signup without meeting all the password requirements	password	Error message displayed	Error message displayed	pass
1.6	Sing up with password and confirmed password not matching	password	Error message displayed	Error message displayed	pass
1.7	Sing up without entering any data		Error message displayed	Error message displayed	pass
1.8	Signup with all the required information	All the information required to signup	Signup successfully	Signup successfully	pass
Test Scenario 2: Check the Sign-in Functionality					
2.2	Sign in with wrong email or password	Email and Password	unable to successfully log in	unable to successfully log in	pass
2.3	Sign in with correct email and password	Email and Password	Sign in successfully	Sign in successfully	pass
Test Scenario 3: Header & Footer Functionalities					
3.1	Click on "الصفحة الرئيسية "		Navigate to homepage	Navigate to homepage	pass
3.2	Click on "من نحن "		Navigate to who we are section at homepage	Navigate to who we are section at homepage	pass
3.3	Click on "خدماتنا "		Navigate to our services section at homepage	Navigate to our services section at homepage	pass
3.4	Click on "تواصل معنا "		Navigate to contact with us section at homepage	Navigate to contact with us section at homepage	pass
3.5	Click on the profile icon		Navigate to the profile page	Navigate to the profile page	pass
3.6	Click on "تسجيل الخروج"		Sign out successfully	Sign out successfully	pass
Test Scenario 4: Check Home Page Functionality					
4.1	Click on "أنشئ هويتك التجارية" service		Navigate to create brand identity page	Navigate to create brand identity page	pass

4.2	Click on " أنشئ محتوى لمناسبة قادمة" service		Navigate to generate marketing content for an upcoming event page	Navigate to generate marketing content for an upcoming event page	pass
4.3	Click on " أنشئ محتوى تسويقي جديد" service		Navigate to generate marketing content page	Navigate to generate marketing content page	pass
4.4	Add user's email and the message and click on send	user's email and the message	Send to MOSAWEEQ's mailbox successfully	Send to MOSAWEEQ's mailbox successfully	pass
Test Scenario 5: Check Home Page Functionality					
5.1	Add business category, preferred language and a description of the business and click on "Next" to suggest business names	business category, preferred language, and a description of the business	Generate list of business names successfully	Generate list of business names successfully	pass
5.2	Choose the preferred colors and click on "Next "to generate logos	business category, preferred language, a description of the business, and chosen colors.	Generate list of logos successfully	Generate list of logos successfully	pass
Test Scenario 6: Check Generate Taglines & Images Pages Functionalities					
6.1	Add a description and click on"أنشئ عبارات تسويقية"	Description	Successfully generate taglines based on the description	Successfully generate taglines based on the description	pass
6.2	Add a description and click on"أنشئ اعلان"	Description	Successfully generate images based on the description	Successfully generate images based on the description	pass
6.3	Add a description, upload an image, and click on" أنشئ اعلان"	Description and Image	Successfully generate list of enhanced and prepared images	Successfully generate list of enhanced and prepared images	pass
Test Scenario 7: Check Generate Marketing Content for an Upcoming Event Page Functionality					
7.1	Click on "أنشئ محتوى" on one of the nearest upcoming events		Successfully navigate to generate new marketing content page	Successfully navigate to generate new marketing content page	pass
Test Scenario 8: Check Profile Page Functionality					
8.1	Modify user's information and click on "حفظ التغييرات"	Modify any of the user's information	Successfully save the modified information	Successfully save the modified information	pass

Table 0.2: system test

6.6 Conclusion

Testing ensures software works correctly and reliably. In this chapter, we covered unit testing, integration testing, compatibility testing, and system testing. Unit tests checked individual features like email verification and password validation. Integration tests ensured components like the login system worked together properly. Compatibility tests confirmed the software ran smoothly on different devices and browsers. System tests verified the overall functionality of the application. All tests were successful, confirming MOSAWEEQ is reliable and ready for users.

Chapter 7: Results and Discussion

7.1 Introduction

This chapter provides a comprehensive navigation guide for utilizing MOSAWEEQ system, ensuring users can effectively operate its features. Following the navigation guide, the chapter will present the objectives achieved by the system, demonstrating the system's success in meeting its intended goals. Additionally, this chapter will address the limitations encountered during the development and implementation of MOSAWEEQ, offering a critical analysis of areas for future improvement.

7.2 System Navigation Guide

A comprehensive navigation guide ensures that users can quickly familiarize themselves with the system, reducing the learning curve and minimizing the potential for errors. It serves as avaluable resource for both novice and experienced users, offering step-by-step guidance and troubleshooting tips that enhance overall usability.
Furthermore, a navigation guide contributes to user satisfaction by empowering users to navigate the system with confidence and ease. It supports the achievement of tasks in a more streamlined and efficient manner, thereby improving productivity and user engagement.

7.2.1 Sign In and Sign-Up Pages

To enhance the user experience and ensure full utilization of the system's functionalities, users should sign up by providing their business information, if available, to be added to the database for future reference. Once registered, users can sign in through the sign in page shown in Figure 7.1 at any time to access the system.

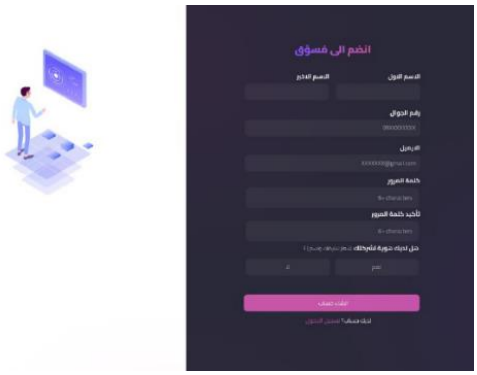


Figure 0.1: Sign Up Page

Figure 0.2: Sign In Page

7.2.2 Home Page

Upon successful login, the user is automatically directed to the homepage, which highlights the three main services. The homepage also includes a brief paragraph explaining what the MOSAWEEQ system is, and the services it offers. Additionally, there is a contact section where users can submit any queries or suggestions that will be directed to MOSAWEEQ's mailbox.

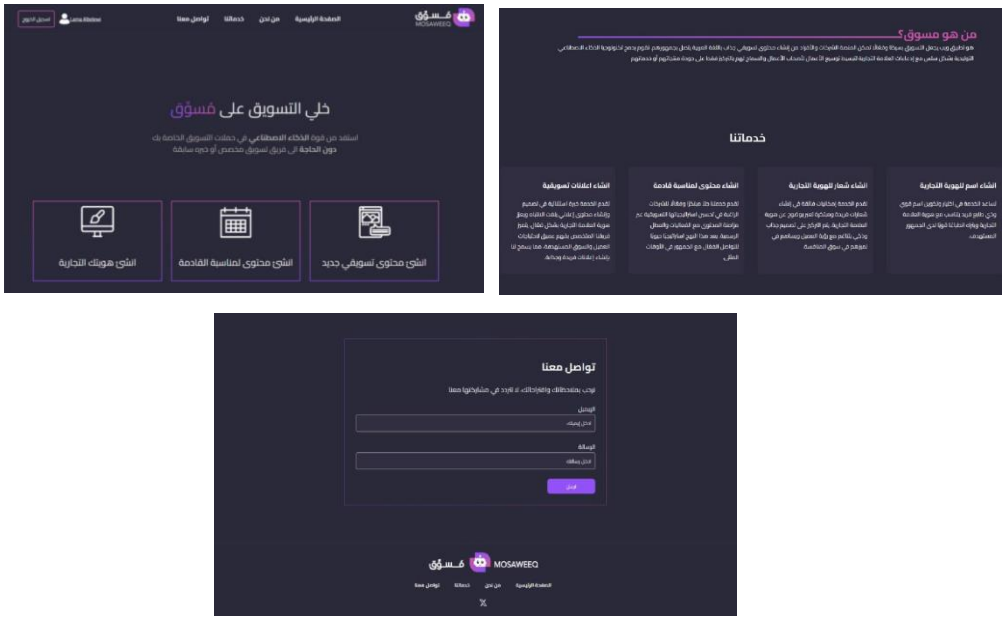


Figure 0.3: Home Page

7.2.3 Create Brand Identity Page

This page allows and assists users in creating a comprehensive and impressive brand identity in minimal time. By entering the business category, selecting the preferred language for the business name suggestions, and providing a description of the business's nature.

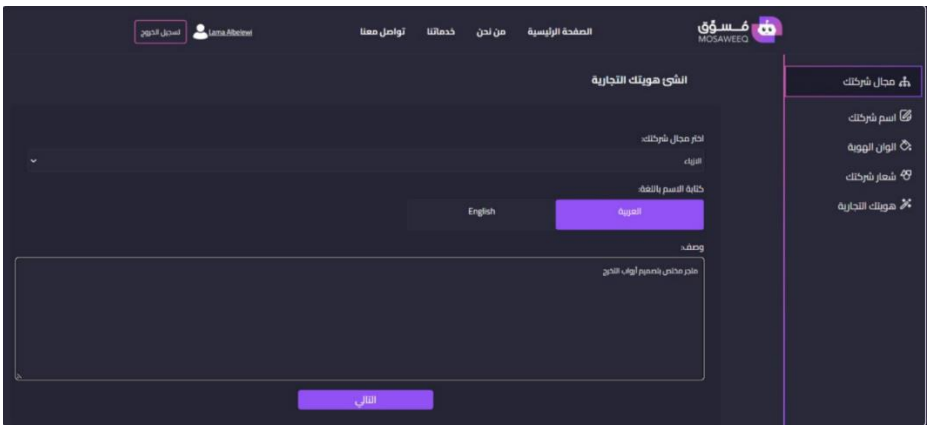


Figure 0.4: Brand Identity page 1

The system generates a list of suggested names along with their meanings based on the previous entries. This step simplifies the process of choosing a suitable name for the business.

Figure 0.5: Brand Identity page 2

After selecting a business name, the user clicks "Next" to proceed with the process by choosing three preferred colors for the brand identity. The user can click on a color to change it, and once the colors are chosen, the user clicks "Next" to continue.

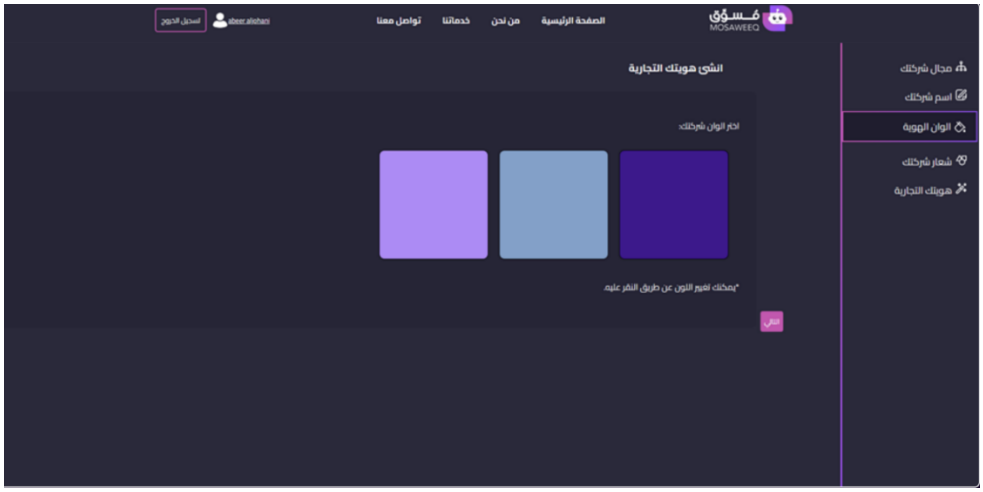


Figure 0.6: Brand Identity page 3

Lastly, six logos are displayed for the user to choose from. Once the preferred logo is selected, the user clicks "Next" to view the complete brand identity, as shown in Figure 7.8

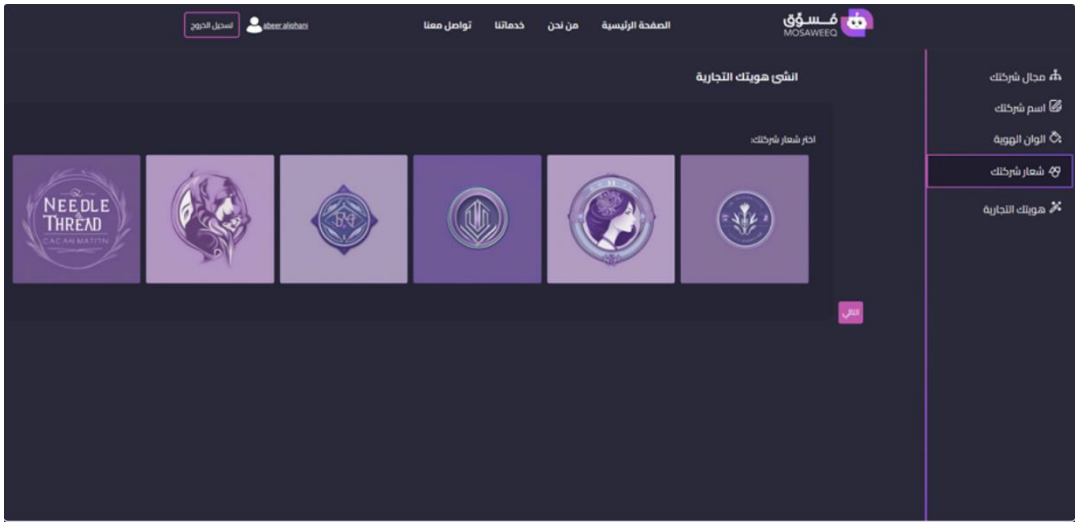


Figure 0.7: Brand Identity page 4

Figure 0.8: Brand Identity page 5

7.2.4 Generate New Marketing Content Page

In this service, users can generate a marketing tagline or image by first selecting a brand from their brand list, which, as shown in image 7.2, contains 8 brands associated with the user.

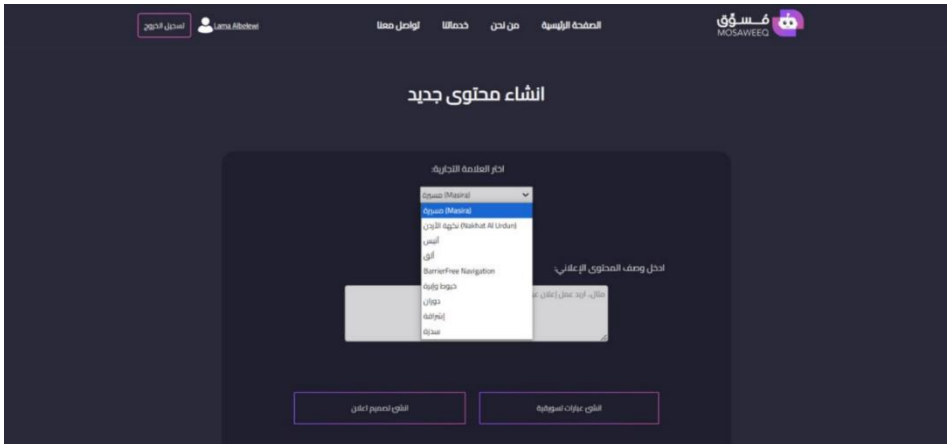


Figure 0.9: generate marketing content page

After that, the user provides a description of the marketing content to be generated and chooses whether to create a marketing tagline or image.

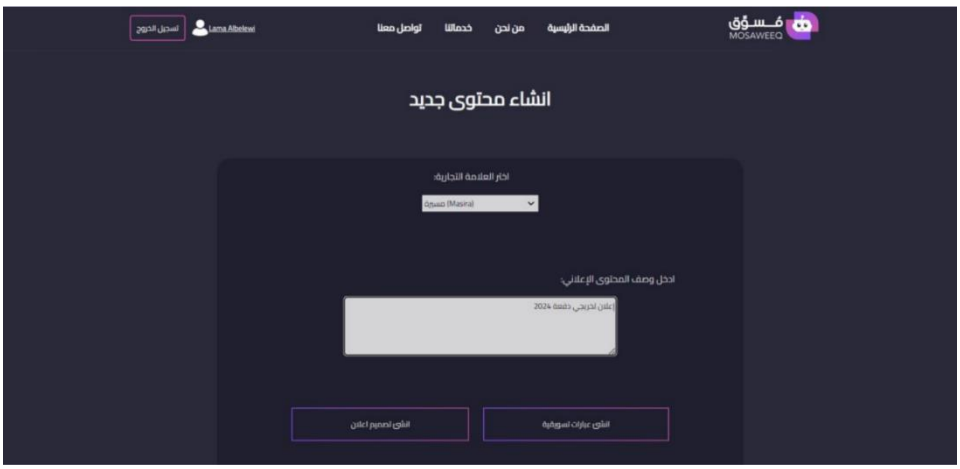


Figure 0.10: generate marketing content page

A marketing tagline will be generated based on the provided description using prompt engineering in the internal layer, ensuring the best results tailored to the user's brand.

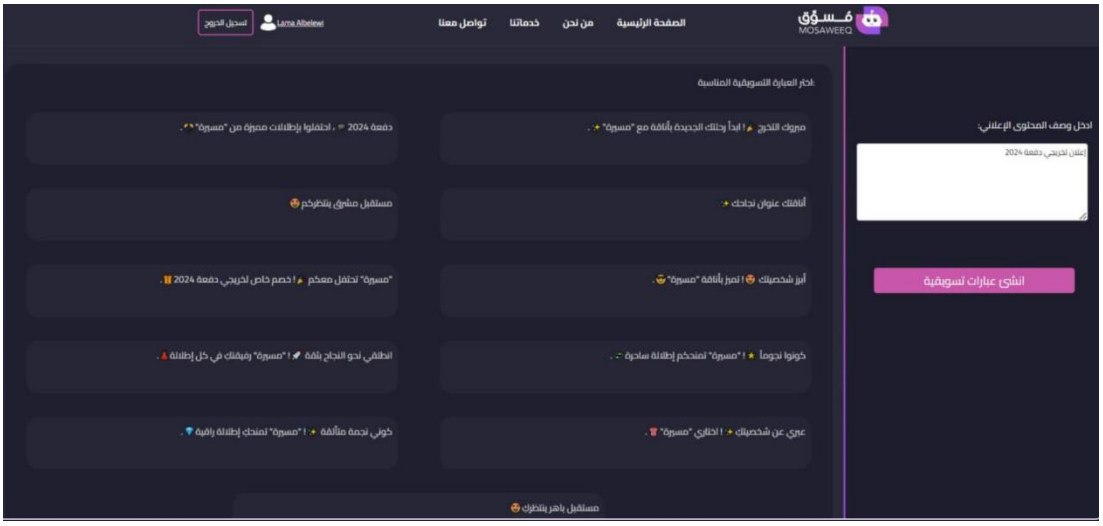


Figure 0.11: marketing tagline page

The marketing image generation sub-service also takes the description and generates marketing images based on it.

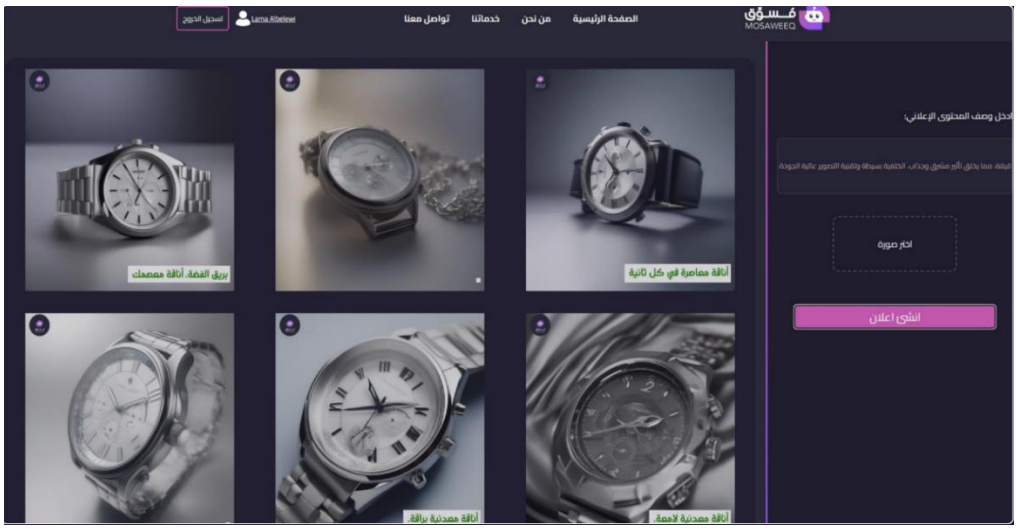


Figure 0.12: marketing image page

Users can upload a product image and describe the desired background in the description field to enhance the background and create a social media-ready image. By clicking on the image and then on 'Save,' users can download the image to their device.

Figure 0.13: marketing image page

7.2.5 Generate Content for an Upcoming Event Page

This service displays a calendar featuring Saudi occasions and Islamic events such as Eid al-Adha and Eid al-Fitr, simplifying the marketing planning process for users. Additionally, it highlights the nearest upcoming events. By clicking on "Generate Content" for one of these upcoming events, user is directed to the generate marketing content page. There, user can add a description for the content to be generated, which will be specifically related to the chosen event.

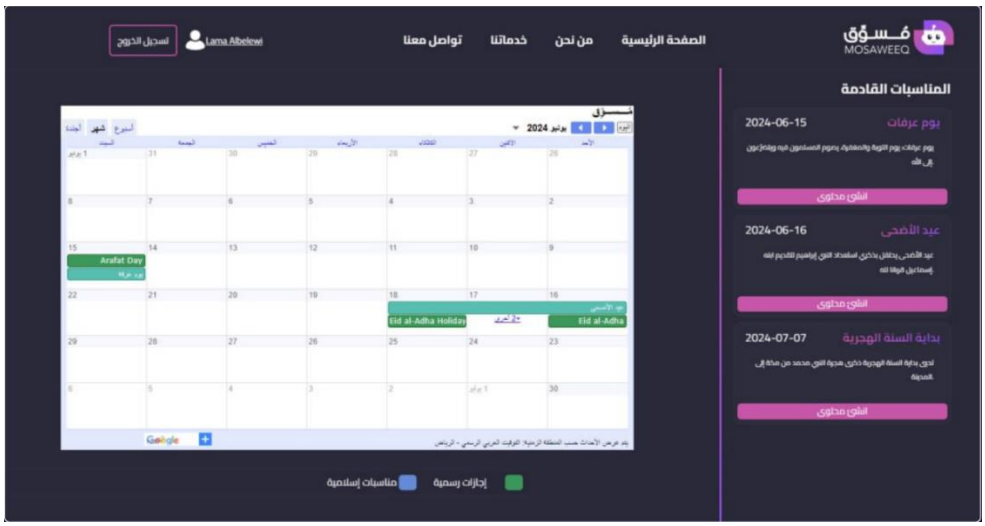


Figure 0.14: Generate Content for an Upcoming Event Page

7.2.6 Profile Page

The profile page showcases the information provided by the user during the signup process. Users have the option to modify and save this information as needed.

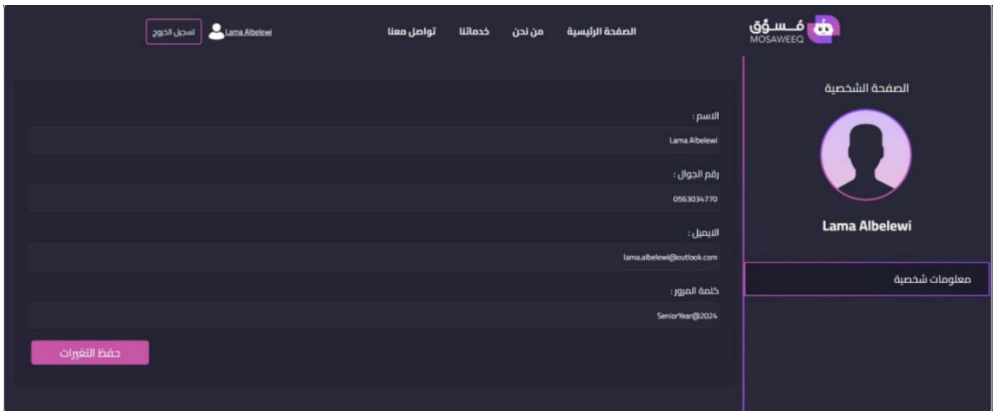


Figure 0.15: Profile page

7.3 Achieved Objectives

MOSAWEEQ project embarked on the development of a web-based application with the primary aim of empowering business owners and individuals to initiate and expand their ventures by simplifying the marketing process and facilitating the creation of compelling marketing campaigns through generative artificial intelligence (AI). Over the course of implementation, the project has successfully achieved all its stated objectives. Through diligent efforts and strategic planning, MOSAWEEQ has effectively empowered productive families and Small and Medium-Sized Enterprises (SMEs) to launch impactful marketing campaigns. By providing accessible tools for crafting engaging marketing content, establishing robust brand identities, and fostering customer engagement through timely and relevant content aligned with special events, MOSAWEEQ has realized its objective of enabling businesses to thrive in the digital landscape. The project's comprehensive approach has not only simplified the marketing process but has also facilitated meaningful connections between businesses and their target audiences, driving growth and success for users across various industries.

7.4 Limitations of the Project

- **Desktop Website Only**: MOSAWEEQ is currently accessible only via desktop browsers, limiting its usability for users who primarily access the internet through mobile devices.
- **Limited Regeneration Options**: Users are unable to regenerate presented logos, marketing images, or taglines directly. In case of generating marketing content, they must modify the description to generate new suggestions.
- **Calendar Event Interaction**: Users cannot click on specific events in the calendar to generate content directly.
- **Delay in Image Loading**: There is a delay in loading images within the system, resulting in all images appearing simultaneously rather than individually. This delay can lead to extended waiting times for users.
- **Unreliable Tagline Generation**: The process for generating marketing taglines sometimes fails to produce complete marketing content, occasionally outputting only the brand name without any additional tagline or context.
- **Inaccurate Image Generation**: The AI model used for generating marketing images often produces images that do not align with the user's description, leading to irrelevant or unusable content.
- **Repetition in Business Name Suggestions**: A list of business names for users aiming to create a brand identity sometimes is repeated, reducing the diversity and usefulness of the suggestions.

7.5 Conclusion

This chapter provides a detailed guide on using the MOSAWEEQ system, from signing up to generating marketing content. The system has successfully achieved its goals by helping business owners, especially small and medium-sized enterprises (SMEs), create effective marketing campaigns and brand identities using AI. Despite these successes, several limitations have been identified that need to be addressed for future improvements. These enhancements will ensure MOSAWEEQ continues to evolve and better meet the needs of its users.

Chapter 8: Conclusion and Future Work

8.1 Conclusion

In conclusion, the development of Mosaweeq has been an incredibly rewarding journey, significantly enhancing our knowledge in web development, AI integration, and API connectivity. Constructing this platform from scratch has provided us with invaluable experience in using new languages and frameworks. Mosaweeq has undergone multiple stages of development to ensure its functionality and effectiveness, successfully simplifying marketing for SMEs. The platform offers comprehensive features that empower businesses to create engaging Arabic marketing content, build strong brand identities, and schedule content for official events and holidays. One of our significant achievements includes the integration of AI models, which enhance the platform's capabilities and user experience. We are confident that Mosaweeq will be a valuable resource for marketers and business owners, supporting their marketing efforts and contributing to their success. Our goal is to make this platform widely available, empowering businesses to streamline their marketing processes and achieve their objectives.

8.2 Future work

Our web application successfully met its primary goals and adhered to our planned timeline. Looking ahead, we aim to incorporate several enhancements to increase the platform's effectiveness and efficiency. The proposed improvements for future development include:

1. **Performance Optimization:** We plan to enhance Mosaweeq's performance to ensure faster processing and response times, improving the overall user experience.
2. **Social Media Integration:** We intend to integrate Mosaweeq with social media APIs, such as Twitter. This integration will enable the analysis of trending topics and commonly used marketing terms, providing users with timely and relevant insights.
3. **Website Launch & Subscription Model:** Our goal is to launch a public website for Mosaweeq and offer it as a subscription-based service, making it accessible to a broader audience.
4. **Enhanced Audience Analysis Features:** We aim to introduce advanced audience analysis tools to Mosaweeq, enabling users to gain deeper insights into their target demographics and optimize their marketing strategies.

By implementing these enhancements, we hope to significantly improve Mosaweeq's capabilities and offer a more robust and user-friendly platform.

Glossary

- 2030 Vision of Saudi Arabia: An initiative outlining the long-term developmental goals and objectives of Saudi Arabia, aiming to diversify the economy, enhance infrastructure, and promote social and economic development.

- Generative artificial intelligence (AI): A type of artificial intelligence that can generate content, such as text, images, and videos, based on provided input or parameters. It is used for various applications in content creation and marketing.

- Marketing content: Text, images, and other materials that are used to promote a product or service.

- Calendar Integration: The ability to link marketing campaigns and content creation with specific dates, events, and holidays for strategic timing.

- Third-Party Services and APIs: External tools or services provided by entities other than the primary software developer, often accessed via Application Programming Interfaces (APIs) to extend functionality or integrate features into a system.

References

[1] "Vision2030," [Online]. Available: https://www.vision2030.gov.sa/en/explore-more/empowering-the-private-sector/.

[2] "Emerging Technologies Adoption," [Online]. Available: https://shorturl.at/qu19X.

[3] C. Kaur and V. Kumar, "Comparative Analysis of Iterative Waterfall," p. 4, March 2015.

[4] "Notion," [Online]. Available: https://www.notion.so/.

[5] Gemini, "Speech-to-Text supported languages," [Online]. Available: https://cloud.google.com/speech-to-text/docs/speech-to-text-supported-languages.

[6] "Huggingface," [Online]. Available: https://huggingface.co/spaces/valhalla/minDALLE/blob/main/CITATION.cff.

[7] Hugging face, "Diffusers Documentation," [Online]. Available: https://huggingface.co/docs/diffusers/index.

[8] "Adcreative," [Online]. Available: https://www.adcreative.ai/.

[9] "Adcreative," [Online]. Available: https://www.adcreative.ai/welcome.

[10] "microsoft Designer," [Online]. Available: https://designer.microsoft.com/.

[11] "BrandCrowd," [Online]. Available: https://shorturl.at/fmUZ6.

[12] "SocialBu," [Online]. Available: https://socialbu.com/.

[13] "Predis.Ai," [Online]. Available: https://predis.ai/.

[14] "predis.ai," [Online]. Available: https://predis.ai/supported-languages/.

[15] E. Suvanto, "Applications of Generative AI in Business," p. 41, 2023.

[16] S. Mayahi and M. Vidrih, "The Impact of Generative AI on the Future of Visual Content Marketing," p. 15.

[17] K. Wiegers, Software Requirements, Microsoft Press, 2013.

[18] "Database Star," [Online]. Available: A Guide to the Entity Relationship Diagram (ERD) - Database Star.

[19] A. H. Bahmani, M. N. and B. Bahmani, "AUTOMATIC DATABASE NORMALIZATION AND PRIMARY KEY GENERATION," p. 16, 2008.

Appendix A

This appendix provides an overview of the selected data-gathering methods utilized in our project, along with a presentation of the obtained results. Our team dedicated extensive efforts to engage with a diverse range of business owners, aiming to ensure the collection of accurate and comprehensive data pivotal for guiding our project's direction.

A.1 Data Gathering Techniques

In the Analysis phase of MOSAWEEQ, a comprehensive approach to data collection was essential to understand the market landscape and the specific needs of potential users. This section outlines the data gathering techniques utilized, primarily involving interview and survey.

A.1.1 Interview

Interviews played a fundamental role in our data collection process, providing us with in-depth insights and perspectives directly from individuals with relevant experience and expertise. The ideation of MOSAWEEQ was sparked by a conversation with a family member of one of our team members, who is a business owner in the embroidery industry. Using her expertise, we conducted structured interviews to gather direct information about the challenges, barriers, and needs faced in marketing an embroidery business.

A.1.1.1 Interview Content

[Name of interviewee: Sara Almarwani] [Interviewee's contact information: @arwab.graduation] [Interview date and location: September 5, 2023| Zoom Meeting] [Your name and position: abeer Aljohani, Lama Abelewi, Rana Alshehri and Jomana Sayadi, development team of MOSAWEEQ]
[Introduction] The following section contains key points from an interview conducted with Sara Almarwani, owner of a small business specializing in graduation gowns. The purpose of this interview was to gain insights into her current marketing campaigns, challenges faced, and areas for potential improvement.
Q1: What are the biggest challenges you face in marketing your business? Sara Almarwani: One of the biggest challenges I face in marketing my business is writing marketing content that is engaging and effective in reaching my target audience. I am not a professional writer or designer, and I do not have the time or resources to create and manage a marketing team. This has limited my ability to create effective marketing campaigns to interact with more customers. Q2: How do you currently create and post your marketing content? Sara Almarwani: I currently create my marketing materials by photographing my graduation gown and sharing the photos on social media. I also communicate with other individuals and organizations in my community to get the word out about my products and services. I have found that word-of-mouth marketing is one of the most effective

ways to reach my target audience. However, I find that these methods can be time-consuming and not always effective in reaching new customers. I would like to find a way to streamline my marketing process and create more effective marketing campaigns.

Q3: What would you like to be able to do differently with your marketing?

Sara Almarwani: I wish I could create marketing campaigns that are customized to my ideal customers. Currently, my marketing efforts lack the attractive touch that I need to capture their attention. I'd love to be able to develop campaigns that speak directly to their needs and interests, using images, and text that resonate with them. This would help me build stronger relationships with my customers and ultimately drive more sales.

Q4: How would you describe your current brand identity?

Sara Almarwani: My current brand identity is somewhat inconsistent. I have a logo and a color palette, but I do not have a clear brand message that communicates what my business is about and what sets me apart from my competitors.

Q5: Have you faced any difficulties in coordinating your marketing efforts with specific events or holidays?

Sara Almarwani: Yes, it has been difficult to coordinate my marketing efforts with specific events or holidays. I often find out about these events too late to plan and execute effective marketing campaigns. This has missed out on opportunities to reach a wider audience and generate sales.

[Your closing comments and overall impression]

Sara Almarwani's insights shed light on the challenges faced by small business owners in creating effective marketing campaigns. Her desire to personalize campaigns and streamline processes demonstrates a deep awareness of potential improvements that could significantly impact her business's success.

A.1.2 Survey

To complement the insights gained from interviews, a survey was designed to obtain a broader perspective from a diverse group of businesses. The survey aimed to gather quantitative data validating the challenges previously identified during the interview section.

The survey was structured to gather information on marketing struggles, brand identity creation, and difficulties in promoting services effectively. By analyzing responses from a larger sample size, the survey outcomes validated the qualitative insights gained from the interview, strengthening the understanding of business-wide issues and needs. This approach provided empirical evidence supporting the necessity for a tool like MOSAWEEQ to assist businesses in overcoming these challenges.

A.2 Questionnaire Questions and Results

In this section, we will show the responses to each question in the questionnaire.

1.1 Are you a business owner?

Asking this question serves to identify the primary target audience. It helps to filter participants: those who are willing to start or already own a business are redirected to relevant sections, while those not involved in business are directed to general questions, ensuring a more customized and relevant interaction for users based on their involvement.

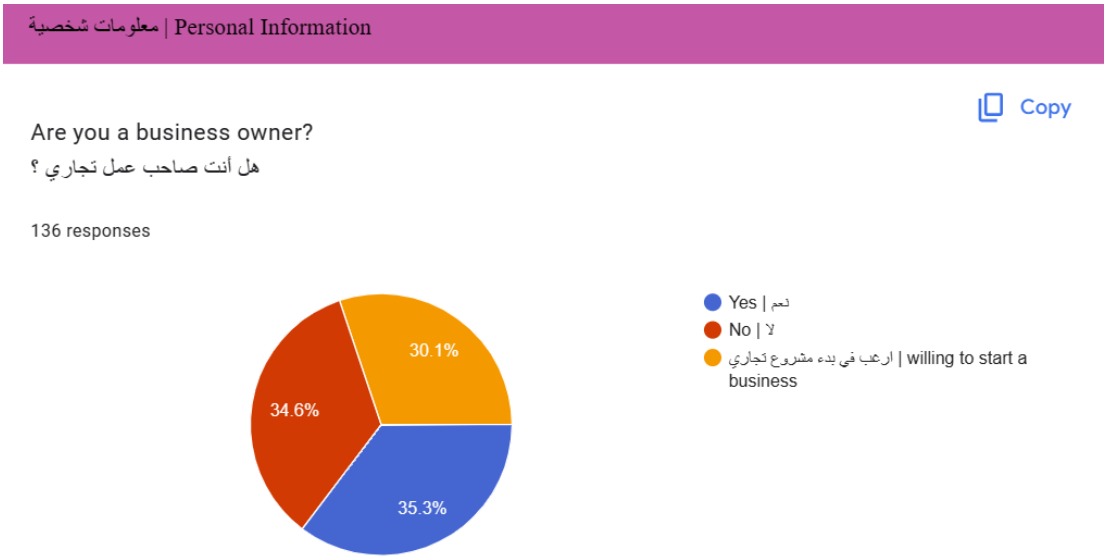


Figure A.1: Question Results

Result: 35.3% of the participants in the questionnaire were business owners, which is the target group for the MOSAWEEQ. Additionally, 30.1% of the participants chose that they are willing to start a business, which is a closely related group for those who are new in business. These two groups together represent 65.4% of the respondents, which is a significant majority.

1.2 What is the scope of your business?

Asking this question helps in understanding the specific needs of the businesses involved. Understanding the scope enables MOSAWEEQ to provide more targeted and relevant assistance, ensuring that business owners receive appropriate tools and guidance aligned with their specific business requirements.

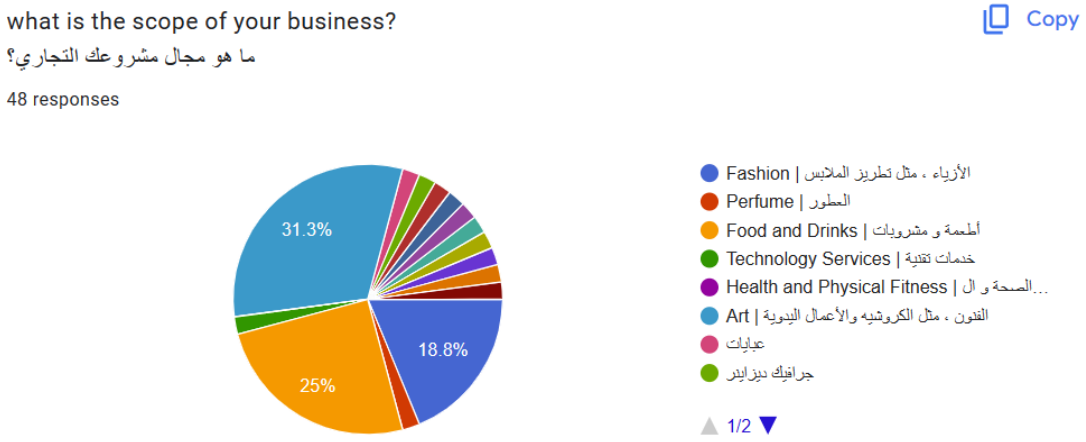


Figure A.2: Question Results

Result: The most popular business sector among questionnaire respondents is art, with 31.3% identifying it as their primary area of business. This is followed by food and drinks at 25%, fashion at 18.8%, with the remaining 21% comprising various sectors such as technology services, health and fitness, and education and others.

1.3 Marketing campaigns in general are expensive compared to the size of most online stores?

Asking this question not only gathers insights into users' perceptions regarding the cost-effectiveness of marketing campaigns but also serves as proof of the necessity for MOSAWEEQ. Understanding this perception allows MOSAWEEQ to customize its offerings by providing cost-effective solutions or tools for smaller online stores. This question helps address affordability challenges, enabling MOSAWEEQ to support businesses in optimizing their marketing campaigns within their budget constraints, thereby proving its relevance and need within the market.

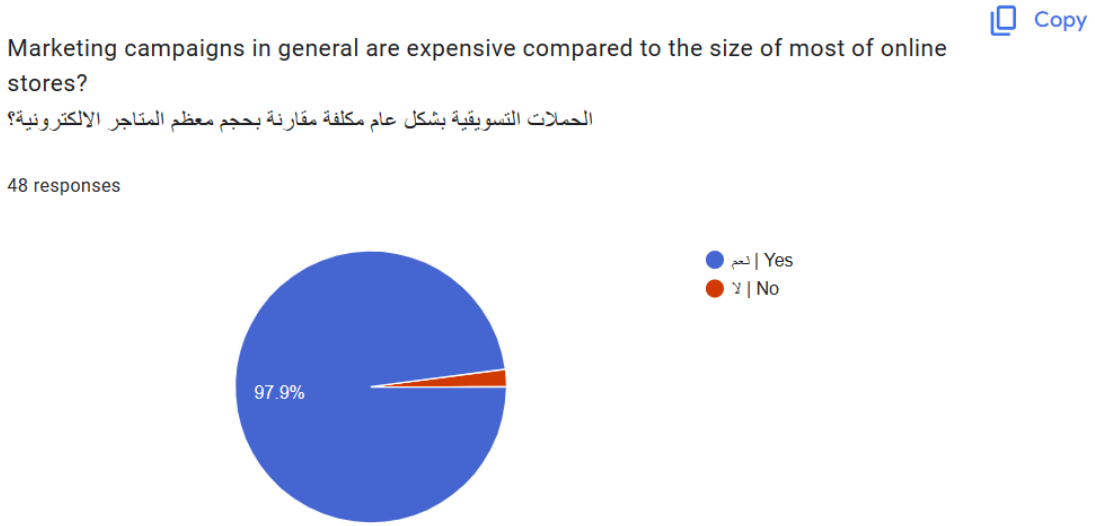


Figure A.3: Question Results

Result: 97.9% of respondents agreed. This means that most online store owners believe that marketing campaigns are too expensive for the size of their stores.

1.4 What is the budget you spent on designing the brand identity?

Asking this question to help MOSAWEEQ assess the financial resources businesses typically allocating or who's willing to allocate towards brand identity design. By understanding the budget range, MOSAWEEQ can customize its offerings and tools to suit various budgetary constraints.

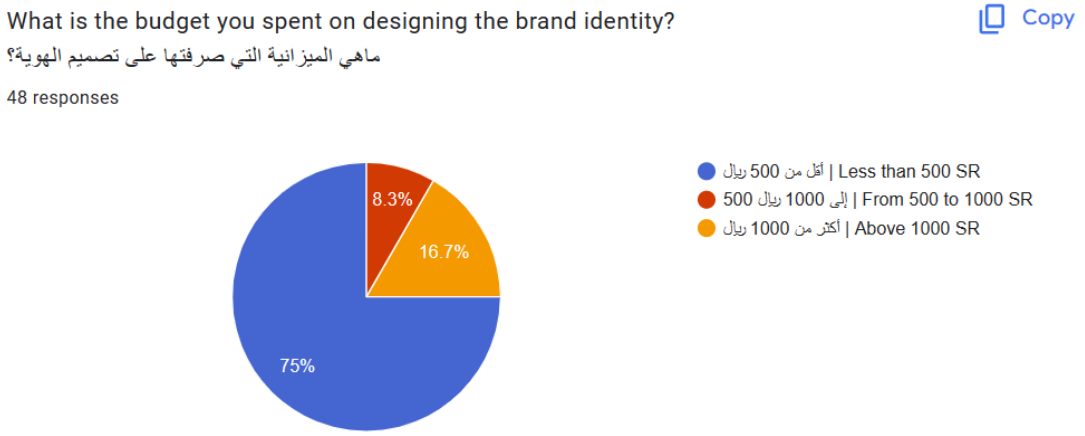


Figure A.4: Question Results

Result: The majority of respondents, comprising 75%, reported spending less than 500 SR on designing their brand identity. A smaller portion, 8.3% of respondents, allocated a budget ranging from 500 to 1000 SR for this purpose. Furthermore, 16.7% of respondents indicated spending above 1000 SR on designing their brand identity. These findings showcase that a significant majority of respondents prefer or have allocated smaller budgets, emphasizing a trend towards cost-effectiveness in brand identity design among the surveyed users.

1.5 I spend an average amount of money (....) per month on marketing campaigns for the store?

Asking this question to understand the budget allocated by business owners towards marketing campaigns for their stores. Knowing the average amount spent per month provides valuable insights to MOSAWEEQ in ensure aligned with the budgetary considerations of businesses, thereby offering more relevant and cost-effective support for their marketing campaigns.

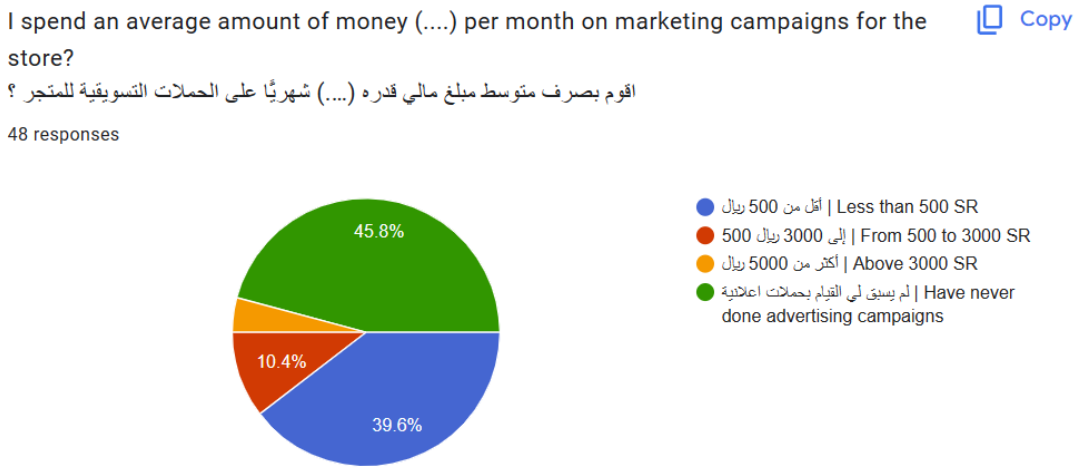


Figure A.5: Question Results

Result: 45.8% of participants indicated that they have never engaged in advertising campaigns for their stores. Around 39.6% reported spending less than 500 Saudi Riyals per month on marketing campaigns. Only 10.4% of respondents spend between 500 and 3000 SR per month on marketing campaigns, and 5.3% spend more than 3000 SR per month. These figures reveal varying budgetary allocations among participants for marketing campaigns, with a significant proportion either not having engaged in advertising campaigns at all or allocating modest budgets for their marketing efforts.

1.6 Have you ever run a marketing campaign that had a significant impact on the store's revenue and profitability?

Asking this question aims to gather insights into the effectiveness of previous marketing efforts for the store owners. Understanding whether their past campaigns have had a tangible impact on sales and profits helps MOSAWEEQ in assessing the success rate of marketing campaigns.

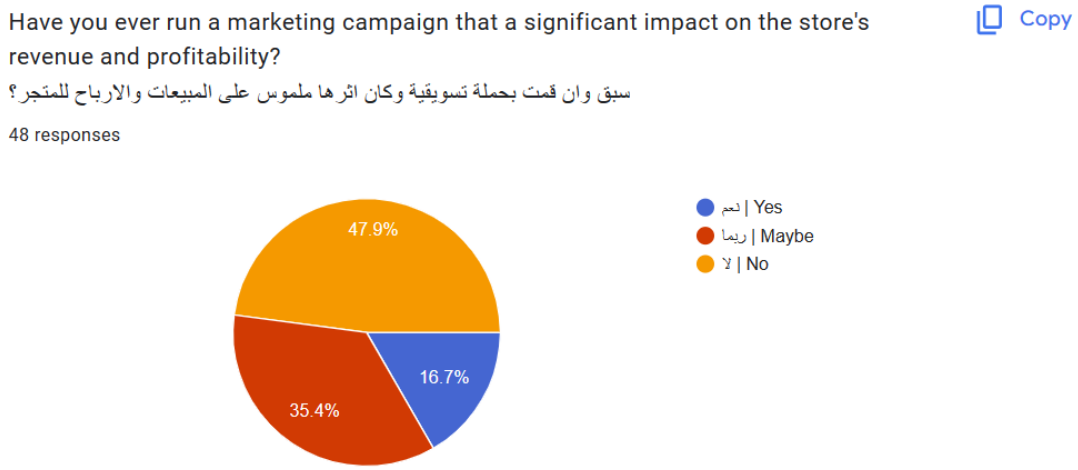


Figure A.6: Question Results

Result: 47.9% of respondents have not experienced a marketing campaign significantly impacting on their store's revenue and profitability. Around 35.4% responded with "Maybe," suggesting uncertainty regarding the impact of their marketing campaigns. Only 16.7% of respondents answered 'Yes,' indicating they have observed a tangible effect on their store's sales and profitability due to a marketing campaign. These results highlight a considerable portion of participants either not observing noticeable impacts from their marketing efforts or expressing uncertainty about the direct influence of marketing campaigns on their store's financial performance.

1.7 How satisfied are you with the quality of your current marketing efforts?

Asking this question serves the purpose of understanding the satisfaction level of business owners regarding their ongoing marketing campaign. This question helps MOSAWEEQ assess the effectiveness and quality of the existing marketing initiatives among users.

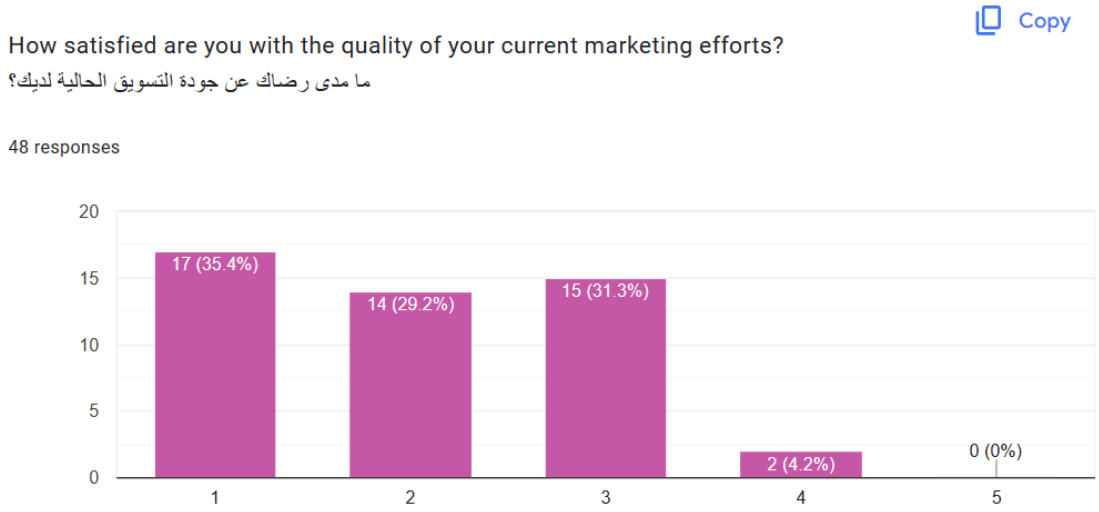


Figure A.7: Question Results

Result: The majority of respondents expressed dissatisfaction with the quality of their current marketing efforts, with a combined total of 64.6% indicating either 'Not Satisfied' (35.4%) or 'Somewhat Unsatisfied' (29.2%). On the other hand, 31.3% of respondents reported a moderate level of satisfaction with their current marketing effort. Notably, there were no respondents who reported being 'Very Satisfied' with their marketing efforts.

1.8 How satisfied are you with the timing of your current marketing efforts? (Do you find them to correspond with official occasions?)

Asking this question aims to gather insights into users' satisfaction with the alignment of their marketing strategies concerning official occasions. This information assists MOSAWEEQ in its objective to provide tools that enhance the effectiveness of marketing campaigns, ensuring they coincide with relevant official events for maximum impact and effectiveness.

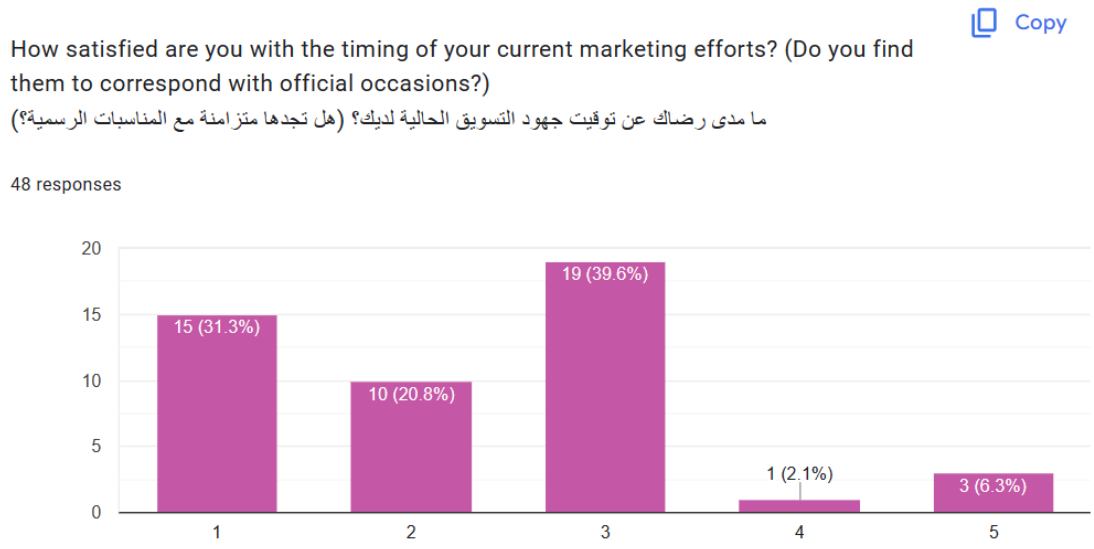


Figure A.8: Question Results

Result: The majority of respondents expressed dissatisfaction with the timing of their current marketing efforts concerning correspondence with official occasions. A combined total of 52.1% of participants indicated 'Not Satisfied' (31.3%) or 'Somewhat Unsatisfied' (20.8%) with the timing alignment of their marketing campaigns. On the other hand, 39.6% reported moderate satisfaction with the timing of their marketing efforts, while only a small percentage, 8.4%, expressed higher satisfaction ('Very Satisfied' or 'Satisfied').

1.9 What are the challenges you face in marketing your products?

Asking this question aims to gain insights into the specific obstacles faced by business owners in their marketing endeavors. Gathering this information not only helps understand the common obstacles encountered by business owners but also represents the necessity for MOSAWEEQ to provide solutions that cater to these challenges.

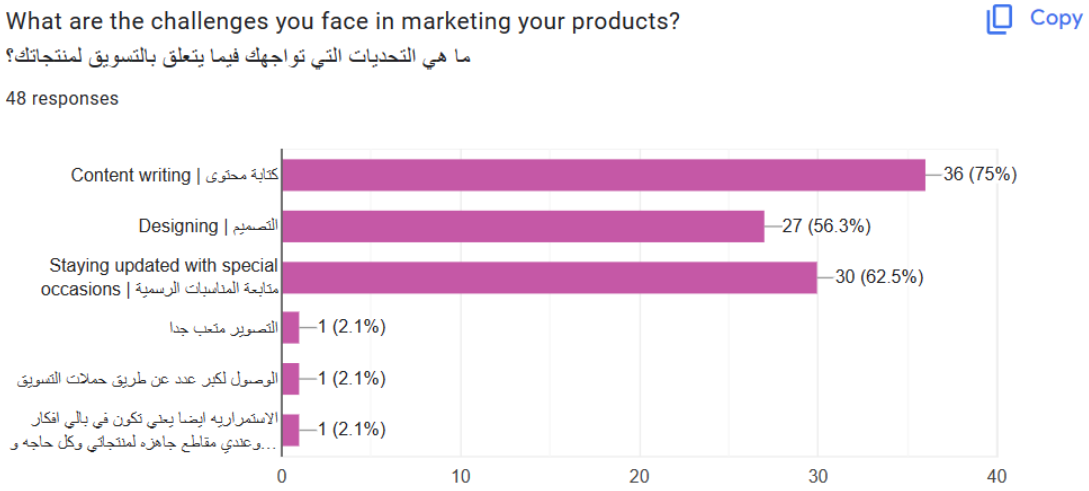


Figure A.9: Question Results

Result: A significant proportion of respondents highlighted various challenges in marketing their products. 75% expressed difficulties in content writing, indicating a need for assistance in generating compelling marketing content. Additionally, 56.3% faced challenges in designing, emphasizing the necessity for support in creating visually appealing materials. Moreover, 62.5% mentioned the difficulty in staying updated with special occasions, highlighting the necessity for tools that assist in keeping up with timely and relevant content updates.

1.10 Do you prefer to have multiple different designs for a single advertising idea to choose the most suitable design for you?

Asking this question to understand users' preferences regarding design diversity for their advertising ideas. This information helps MOSAWEEQ to validate the need to offer multiple design options to cater to users' preferences and enhance their ability to select the most suitable design.

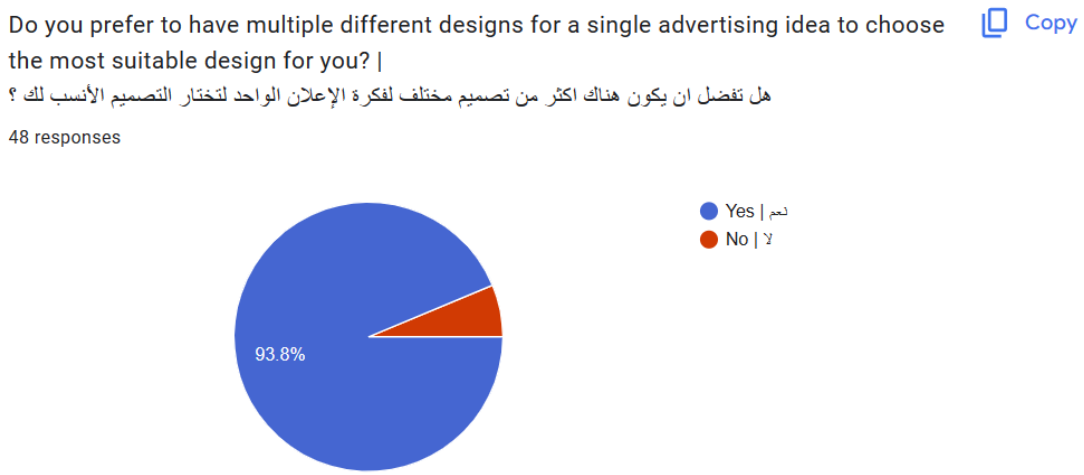


Figure A.10: Question Results

Result: The majority of respondents, comprising 93.8%, indicated a preference for having multiple different designs for a single advertising idea to choose the most suitable design for their needs.

1.11 Do you think AI could be used to improve the cost, quality, timing, and work rate of your marketing efforts?

Asking this question to understand respondents' perceptions and expectations regarding the potential use of artificial intelligence (AI) in enhancing various aspects of their marketing efforts. MOSAWEEQ aims to assess the level of awareness and acceptance of AI technology among users. This information is crucial for MOSAWEEQ to align its services and offerings, emphasizing the benefits of AI for marketing solutions.

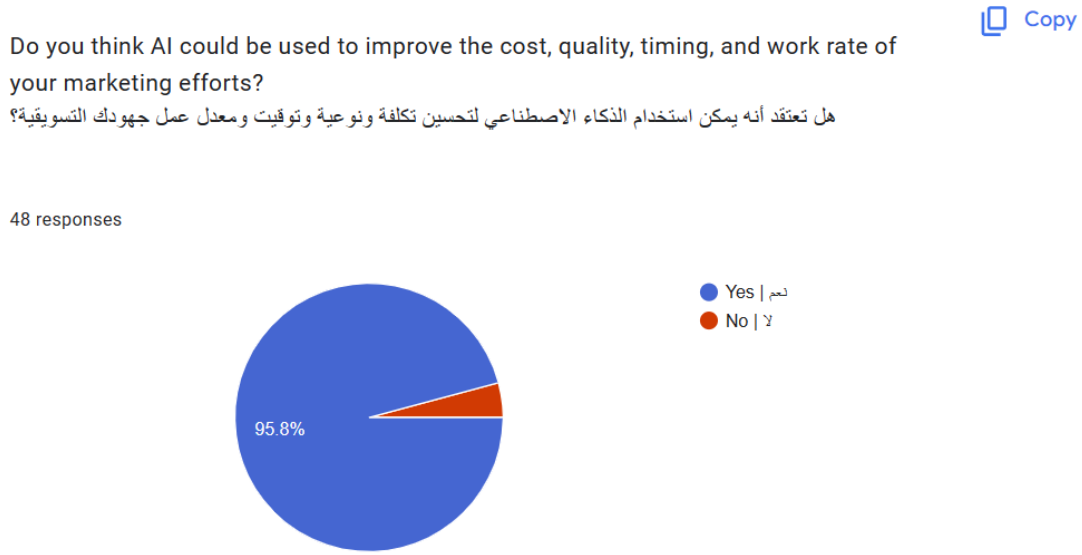


Figure A.11: Question Results

Result: The majority of respondents, comprising 95.8%, believe that AI could be used to improve the cost, quality, timing, and efficiency of their marketing efforts. This result shows that many people recognize the potential benefits of AI for speeding up marketing campaigns.

1.12 What are the challenges you face in starting your new business?

The goal of this question is to gather information about the obstacles faced by aspiring and new business owners so we can focus on them during our project.

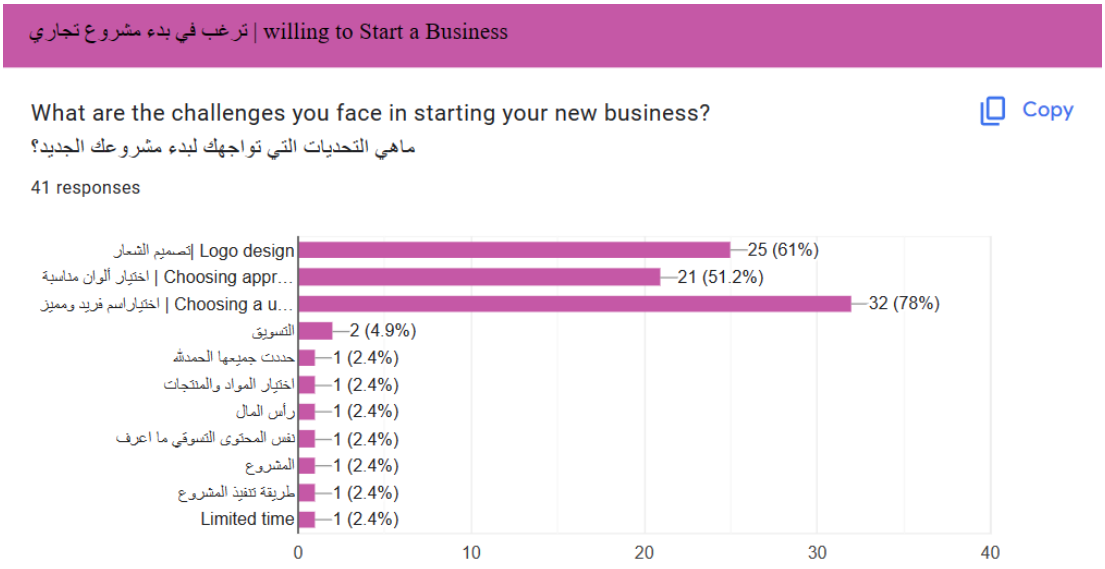


Figure A.12: Question Results

Result: The information resulting from this question helped us to identify common challenges faced by people who are willing to start a new business. Among the most faced challenges are selecting a unique brand name, designing a memorable logo, and choosing a theme from a color template. A survey found that 78% of respondents struggled with brand naming, 61% faced difficulties with logo design, and 51% encountered challenges in selecting a color theme.

1.13 What is the budget allocated for designing the brand identity?

In this question, we aimed to gather essential information on financial resources. This data helps in conducting a cost-benefit analysis for the brand identity design that people can afford.

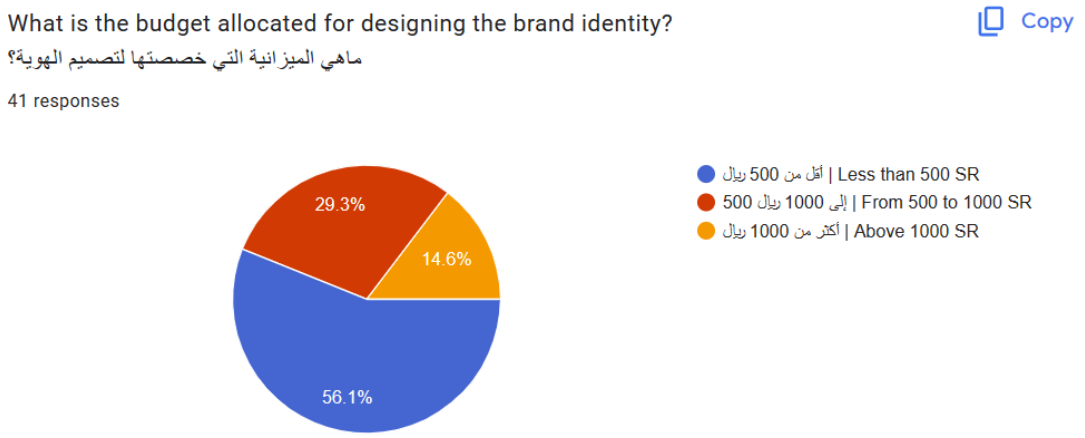


Figure A.13: Question Results

Result: The survey results reveal that a significant portion of respondents, representing 56.1%, allocated a budget of less than 500 SR for their brand identity. This aligns favorably with the cost-effectiveness of utilizing generative AI models or outsourcing services, which typically fall within this budget range. Additionally, 29.3% of respondents invested between 500 and 1000 SR, and 14.6% allocated a budget exceeding 1000 SR.

1.14 What are the factors that affect your satisfaction with the costs of designing the identity for the business?

The goal of this survey question is to identify the factors influencing satisfaction with the costs of designing business identity. It seeks to understand people's expectations and criteria for defining a design as good.

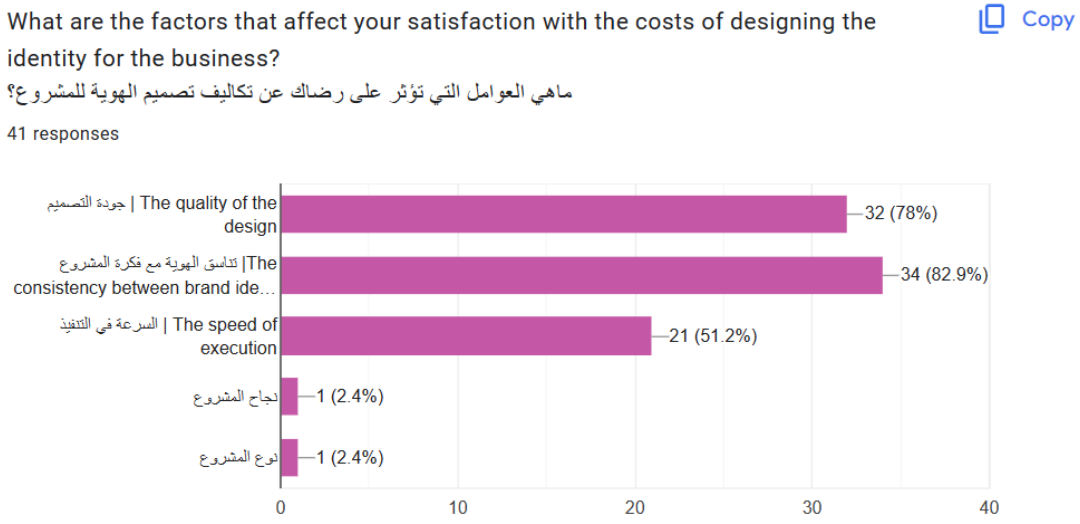


Figure A.14: Question Results

Result: We conclude from this result that the majority 82.9% emphasize consistency between the brand name and its visual components (logo and theme), and 78% of the respondents prioritize high-quality design, seeking a visually appealing and professional brand identity. Furthermore, 51.2% of respondents value prompt completion of the brand identity design, indicating a desire for efficiency and timely delivery.

1.15 Do you think that AI-powered marketing can help achieve the goals of the 2030 vision by solving many of the challenges faced by business owners?

The objective of this question is to gather societal opinions on the alignment of our project idea with the goals outlined in the 2030 vision. It seeks to understand if the public perceives our project as congruent with the broader objectives set forth in the 2030 vision.

Figure A.15: Question Results

Result: 94.9% of individuals perceive that our project aligns well with the objectives outlined in the 2030 vision, which is a high percentage of agreement.

1.16 If I found a tool for creating marketing content for my store, this will be a suitable option for me to rely on during marketing campaigns.

The aim of this question is to gather individuals' opinions on whether they would rely on a tool for creating marketing content if such a tool existed. The focus is on understanding if the task is challenging, especially for small businesses, and whether the availability of a content creation tool would influence their reliance on it.

Figure A.16: Question Results

Result: The substantial agreement of 98.8% among individuals indicates a strong inclination to rely on the tool, providing heightened assurance in the project's practicality and appeal.

A.3 Blank Copy of the Survey

معلومات شخصية | Personal Information

Are you a business owner? *
هل أنت صاحب عمل تجاري ؟

- Yes | نعم
- No | لا
- willing to start a business | ارغب في بدء مشروع تجاري

Figure A.17: Question 1

صاحب عمل تجاري | a Business Owner

Age: *
العمر:

- Under | أقل
- 18-28
- 29-39
- Above | أعلى

Figure A.18: Question 2

what is the scope of your business? *
ما هو مجال مشروعك التجاري؟

- ◯ Fashion | الأزياء ، مثل تطريز الملابس
- ◯ Perfume | العطور
- ◯ Food and Drinks | أطعمة و مشروبات
- ◯ Technology Services | خدمات تقنية
- ◯ Health and Physical Fitness | الصحة و اللياقة البدنية
- ◯ Art | الفنون ، مثل الكروشيه والأعمال اليدوية
- ◯ Other:

Figure A.19: Question 3

What is your Business' Instagram account?
ما هو حساب انستقرام الخاص بمشروعك التجاري؟

Your answer

Figure A.20: Question 4

Marketing campaigns in general are expensive compared to the size of most of online stores? *
الحملات التسويقية بشكل عام مكلفة مقارنة بحجم معظم المتاجر الالكترونية؟

- ◯ Yes | نعم
- ◯ No | لا

Figure A.21: Question 5

What is the budget you spent on designing the brand identity? *
ماهي الميزانية التي صرفتها على تصميم الهوية؟

- Less than 500 SR | أقل من 500 ريال
- From 500 to 1000 SR | 500 إلى 1000 ريال
- Above 1000 SR | أكثر من 1000 ريال

Figure A.22: Question 6

I spend an average amount of money (....) per month on marketing campaigns for the store? *
اقوم بصرف متوسط مبلغ مالي قدره (....) شهريًّا على الحملات التسويقية للمتجر ؟

- Less than 500 SR | أقل من 500 ريال
- From 500 to 3000 SR | 500 إلى 3000 ريال
- Above 3000 SR | أكثر من 5000 ريال
- Have never done advertising campaigns | لم يسبق لي القيام بحملات اعلانية

Figure A.23: Question 7

Have you ever run a marketing campaign that a significant impact on the store's revenue and profitability? *
سبق وان قمت بحملة تسويقية وكان اثرها ملموس على المبيعات والارباح للمتجر؟

- Yes | نعم
- Maybe | ربما
- No | لا

Figure A.24: Question 8

How satisfied are you with the quality of your current marketing efforts? *
ما مدى رضاك عن جودة التسويق الحالية لديك؟

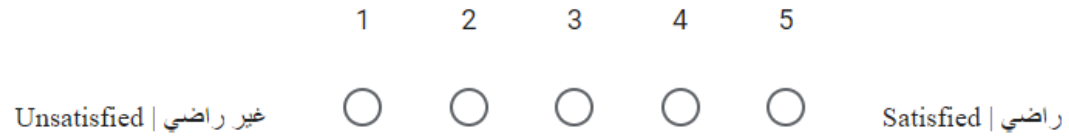


Figure A.25: Question 9

How satisfied are you with the timing of your current marketing efforts? (Do you find them to correspond with official occasions?) *
ما مدى رضاك عن توقيت جهود التسويق الحالية لديك؟ (هل تجدها متزامنة مع المناسبات الرسمية؟)

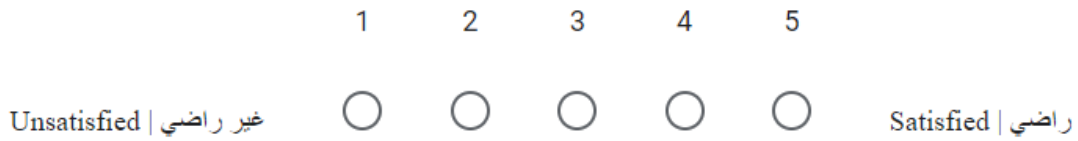


Figure A.26: Question 10

What are the challenges you face in marketing your products? *
ما هي التحديات التي تواجهك فيما يتعلق بالتسويق لمنتجاتك؟

- [ ] Content writing | كتابة محتوى
- [ ] Designing | التصميم
- [ ] Staying updated with special occasions | متابعة المناسبات الرسمية
- [ ] Other:

Figure A.27: Question 11

Do you prefer to have multiple different designs for a single advertising idea to choose the most suitable design for you? | *
هل تفضل ان يكون هناك اكثر من تصميم مختلف لفكرة الإعلان الواحد لتختار التصميم الأنسب لك ؟

- ◯ Yes | نعم
- ◯ No | لا

Figure A.28: Question 12

Do you think AI could be used to improve the cost, quality, timing, and work rate of your marketing efforts? *
هل تعتقد أنه يمكن استخدام الذكاء الاصطناعي لتحسين تكلفة ونوعية وتوقيت ومعدل عمل جهودك التسويقية؟

- ◯ Yes | نعم
- ◯ No | لا

Figure A.29: Question 13

ترغب في بدء مشروع تجاري | willing to Start a Business

What are the challenges you face in starting your new business? *
ماهي التحديات التي تواجهك لبدء مشروعك الجديد؟

- ☐ Logo design | تصميم الشعار
- ☐ Choosing appropriate colors | اختيار ألوان مناسبة
- ☐ Choosing a unique and distinctive name | اختياراسم فريد ومميز
- ☐ Other: _____

Figure A.30: Question 14

What is the budget allocated for designing the brand identity? *
ماهي الميزانية التي خصصتها لتصميم الهوية؟

- ◯ Less than 500 SR | أقل من 500 ريال
- ◯ From 500 to 1000 SR | 500 إلى 1000 ريال
- ◯ Above 1000 SR | أكثر من 1000 ريال

Figure A.31: Question 15

What are the factors that affect your satisfaction with the costs of designing the identity for the business? *
ماهي العوامل التي تؤثر على رضاك عن تكاليف تصميم الهوية للمشروع؟

- ☐ The quality of the design | جودة التصميم
- ☐ The consistency between brand identity and business idea| تناسق الهوية مع فكرة المشروع
- ☐ The speed of execution | السرعة في التنفيذ
- ☐ Other: _____

Figure A.32: Question 16

معلومات عامة | General Information

Do you think that AI-powered marketing can help achieve the goals of the 2030 vision by solving many of the challenges faced by business owners? *
هل تعتقد أن التسويق المدعوم بالذكاء الاصطناعي يمكن أن يساعد في تحقيق أهداف رؤية 2030 من خلال حل العديد من التحديات التي يواجهها أصحاب الأعمال؟

- ◯ Yes | نعم
- ◯ No | لا

Figure A.33: Question 17

If I found a tool for creating marketing content for my store, this will be a suitable option for me to rely on during marketing campaigns. *
إن وجدت أداة لصنع المحتوى التسويقي لمنشأتي/ متجري، سيعد ذلك خيارا مناسبا لي للاعتماد عليه خلال الحملات التسويقية

- ◯ Yes | نعم
- ◯ No | لا

Figure A.34: Question 18

Do you have any comments, suggestions, or concerns regarding our project idea?
هل لديك أية تعليقات أو اقتراحات تود مشاركتها بخصوص فكرة مشروعنا؟

Your answer

Figure A.35: Question 19

If you have any questions or concern, please provide your contact number or email:
إذا كان لديك أي أسئلة أو استفسارات، يرجى تقديم رقم هاتفك أو بريدك الإلكتروني:

Your answer

Figure A.36: Question 20

A.4 Survey Detailed Results

Personal Information | معلومات شخصية

?Are you a business owner
هل أنت صاحب عمل تجاري ؟

136 ردًا

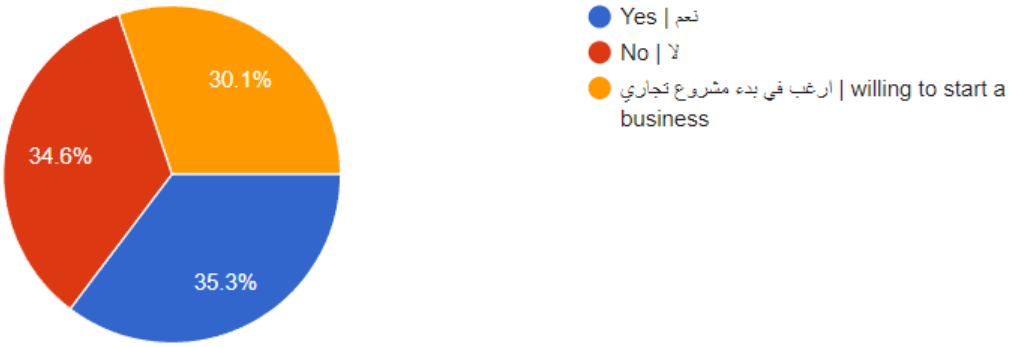


Figure A.37: Question 1 Results

a Business Owner | صاحب عمل تجاري

:Age
:العمر

105 ردود

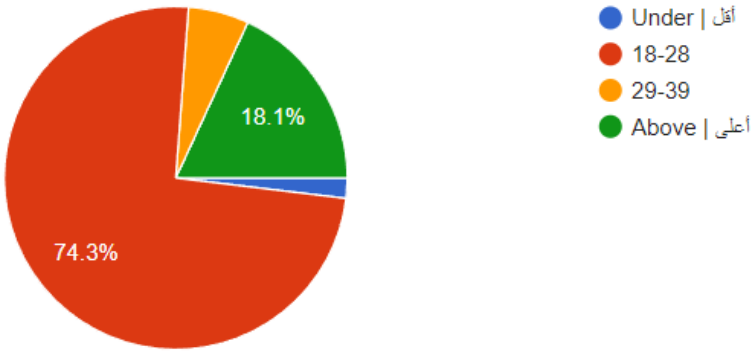


Figure A.38: Question 2 Results

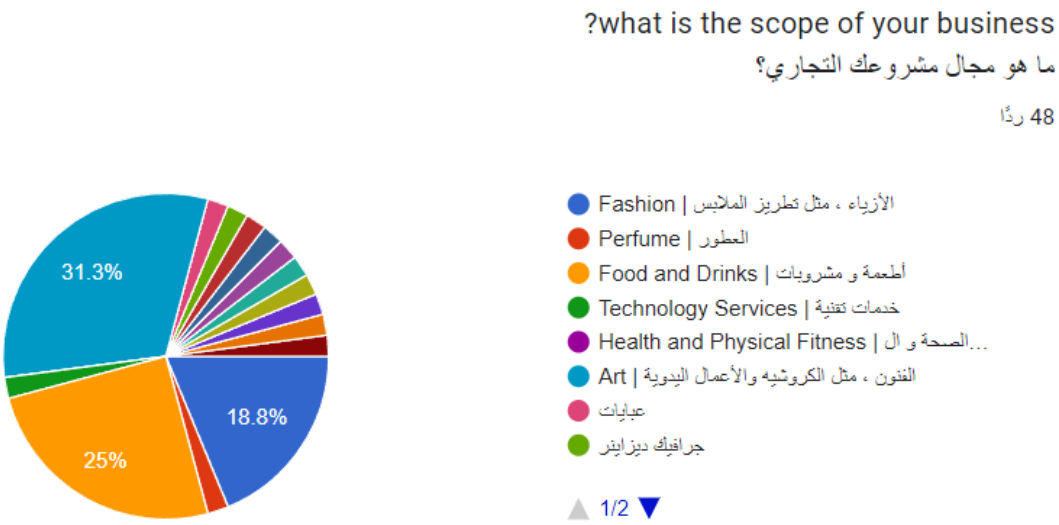


Figure A.39: Question 3 Results

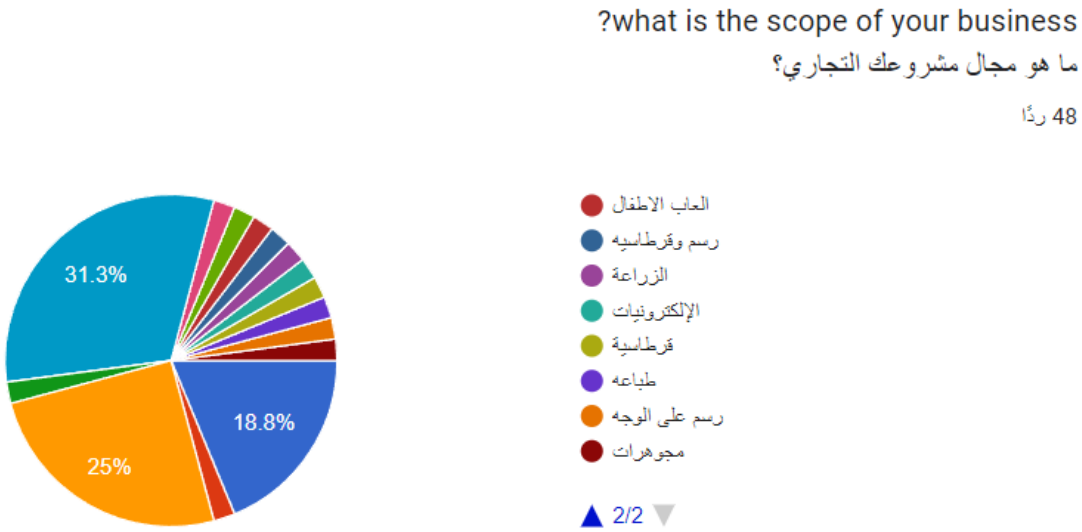


Figure A.40: Question 3 Results

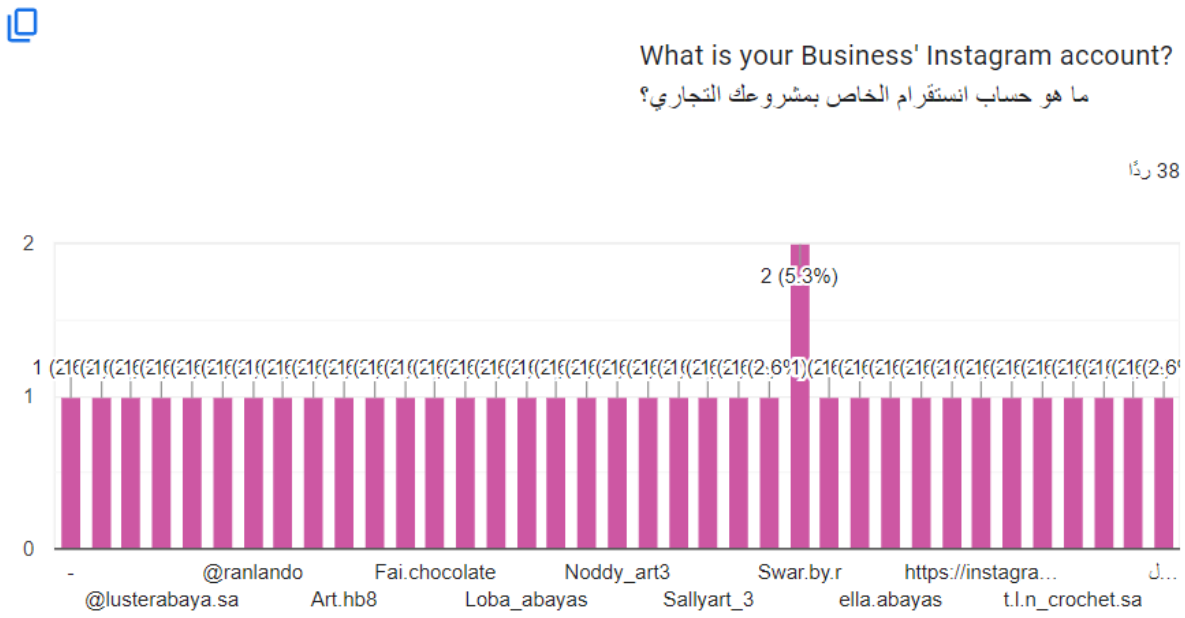


Figure A.41: Question 4 Results

Figure A.42: Question 5 Results

?What is the budget you spent on designing the brand identity

ماهي الميزانية التي صرفتها على تصميم الهوية؟

48 ردًا

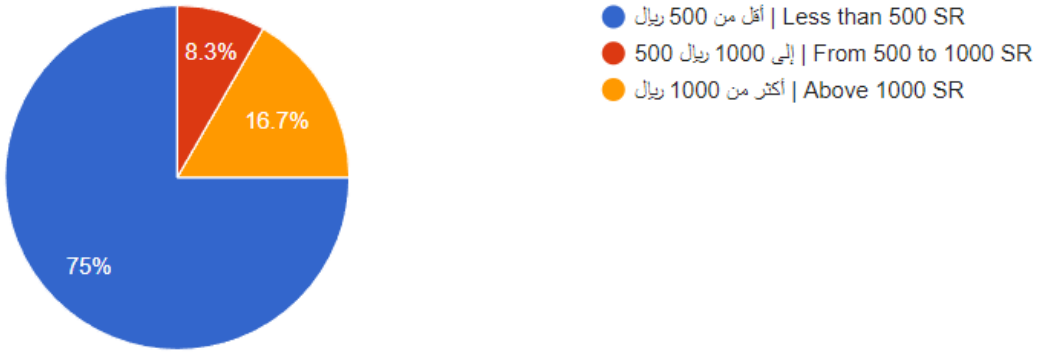


Figure A.43: Question 6 Results

I spend an average amount of money (....) per month on marketing campaigns for the store?

اقوم بصرف متوسط مبلغ مالي قدره (....) شهريًا على الحملات التسويقية للمتجر ؟

48 ردًا

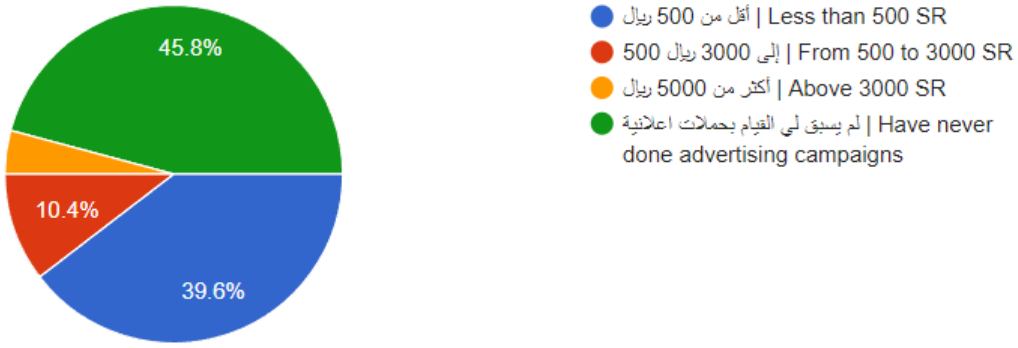


Figure A.44: Question 7 Results

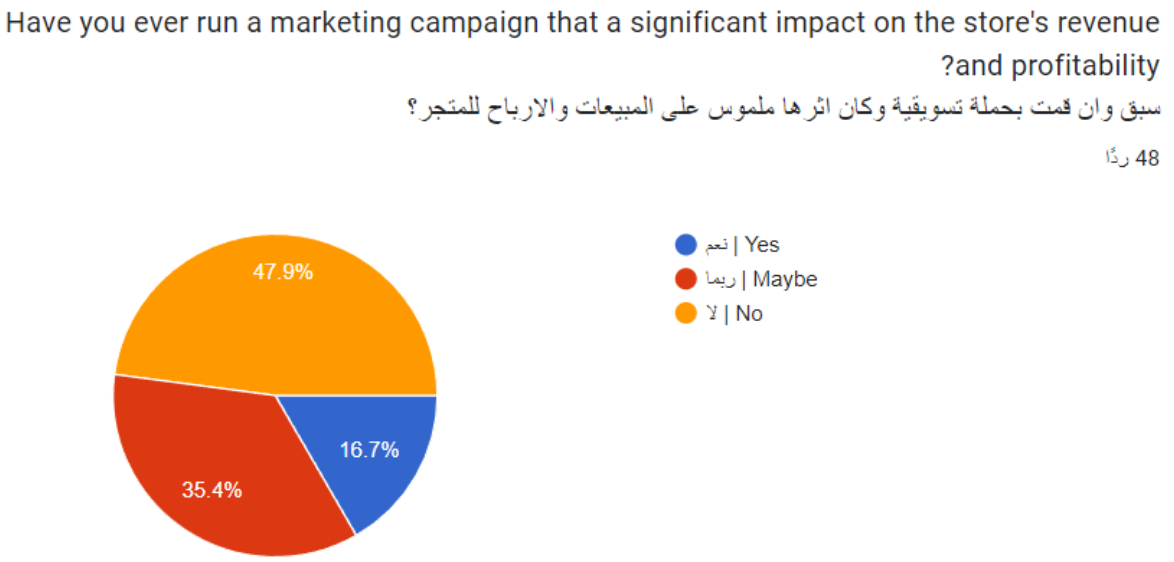


Figure A.45: Question 8 Results

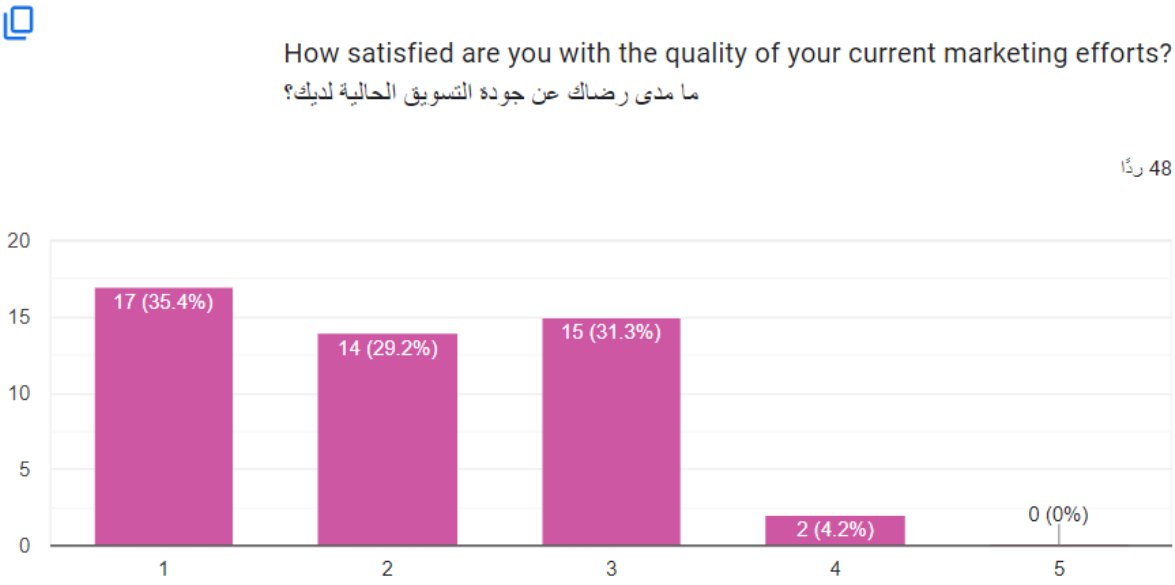


Figure A.46: Question 9 Results

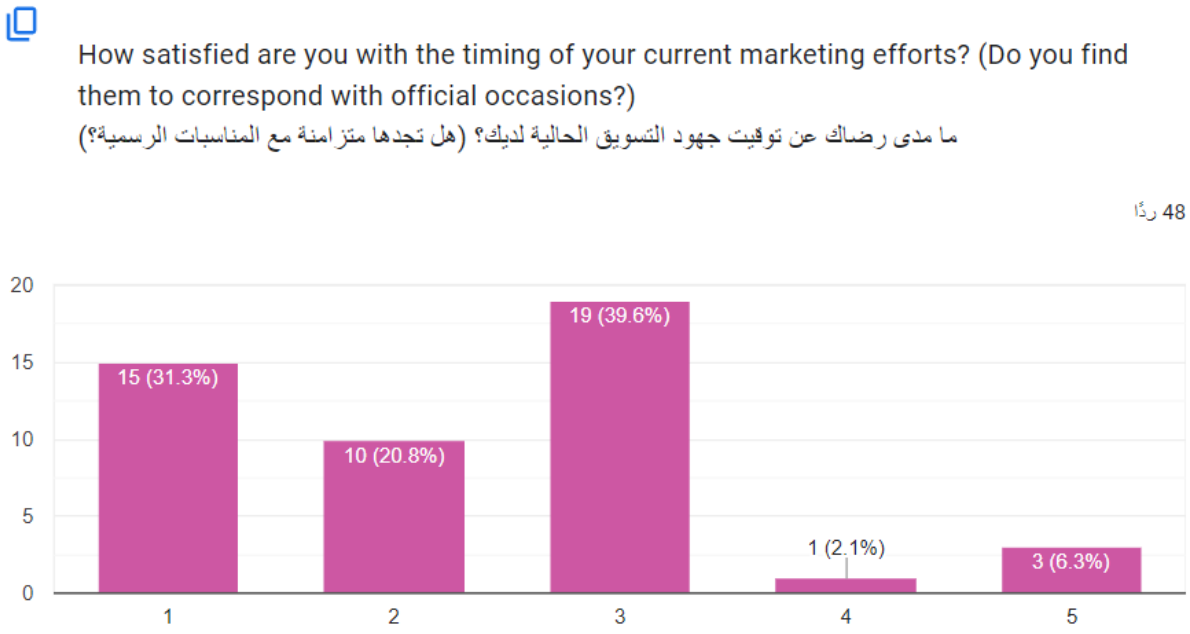


Figure A.47: Question 10 Results

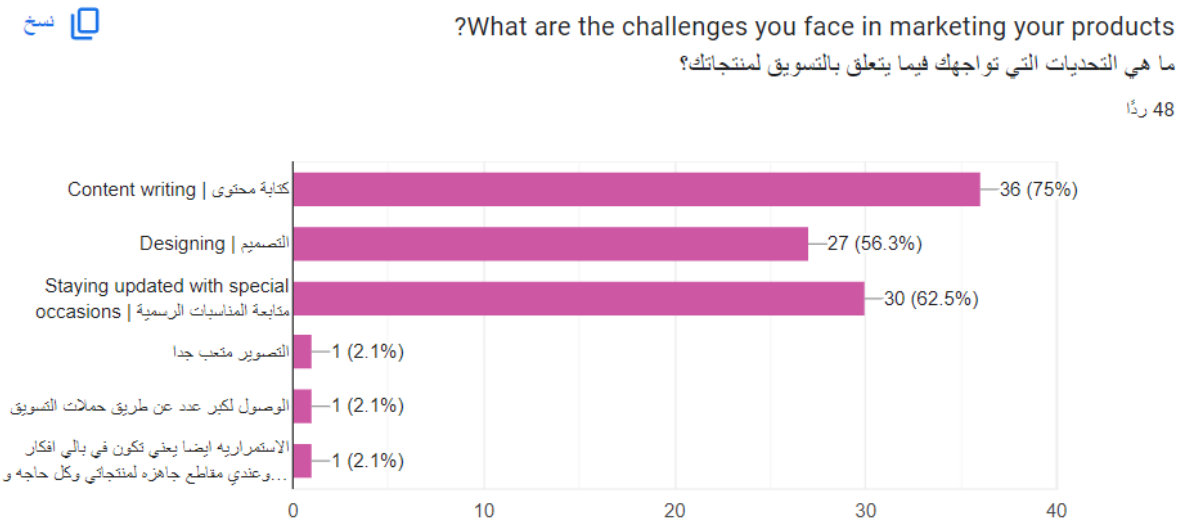


Figure A.48: Question 11 Results

Do you prefer to have multiple different designs for a single advertising idea to choose | ?the most suitable design for you

هل تفضل ان يكون هناك اكثر من تصميم مختلف لفكرة الإعلان الواحد لتختار التصميم الأنسب لك ؟

48 ردًا

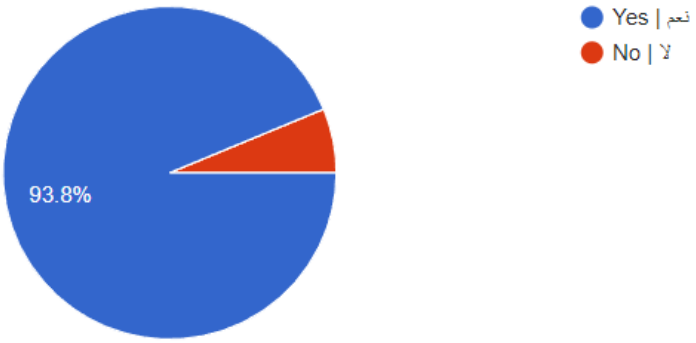


Figure A.49: Question 12 Results

Do you think AI could be used to improve the cost, quality, timing, and work rate of your marketing efforts?

هل تعتقد أنه يمكن استخدام الذكاء الاصطناعي لتحسين تكلفة ونوعية وتوقيت ومعدل عمل جهودك التسويقية؟

48 ردًا

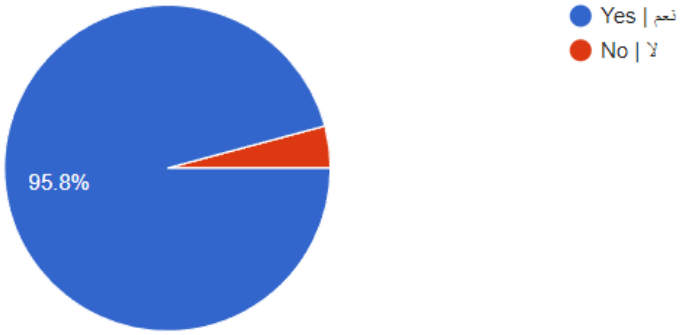


Figure A.50: Question 13 Results

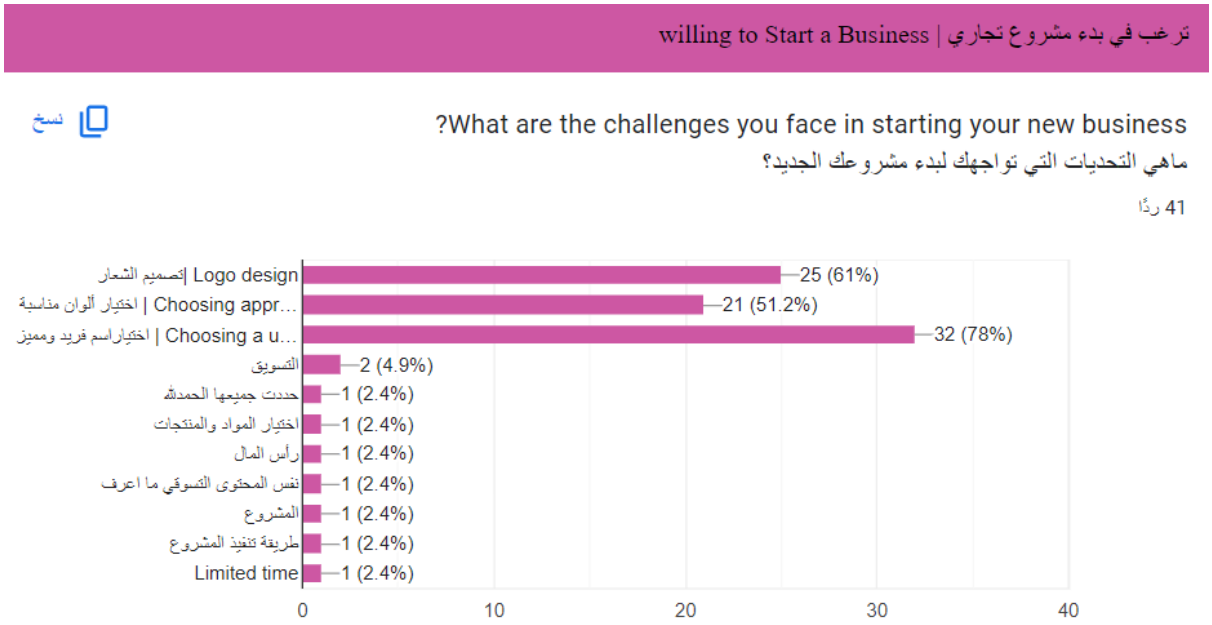


Figure A.51: Question 14 Results

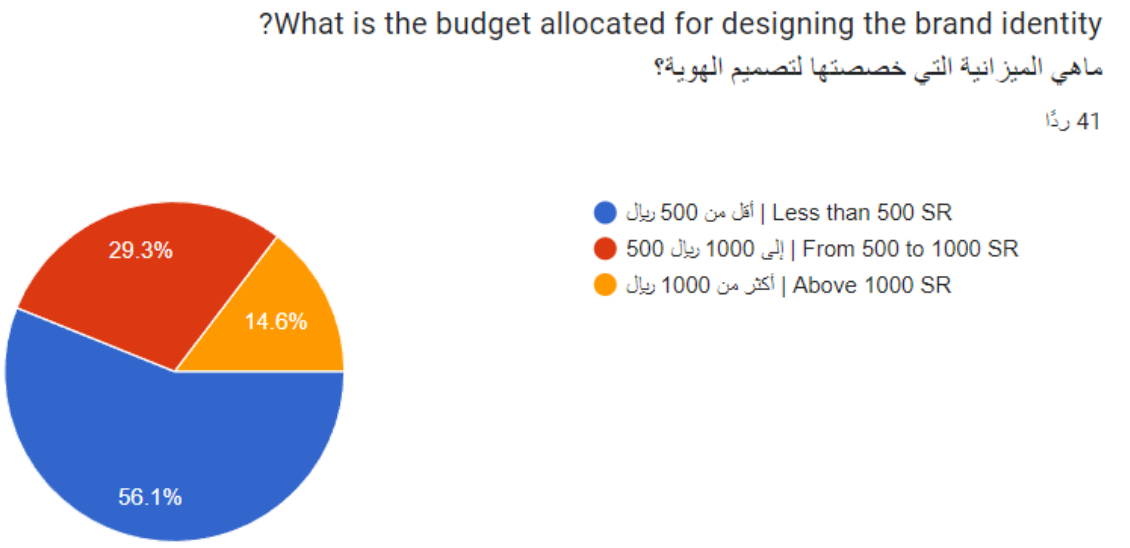


Figure A.52: Question 15 Results

نسخ

What are the factors that affect your satisfaction with the costs of designing the identity for the business?

ماهي العوامل التي تؤثر على رضاك عن تكاليف تصميم الهوية للمشروع؟

41 ردًا

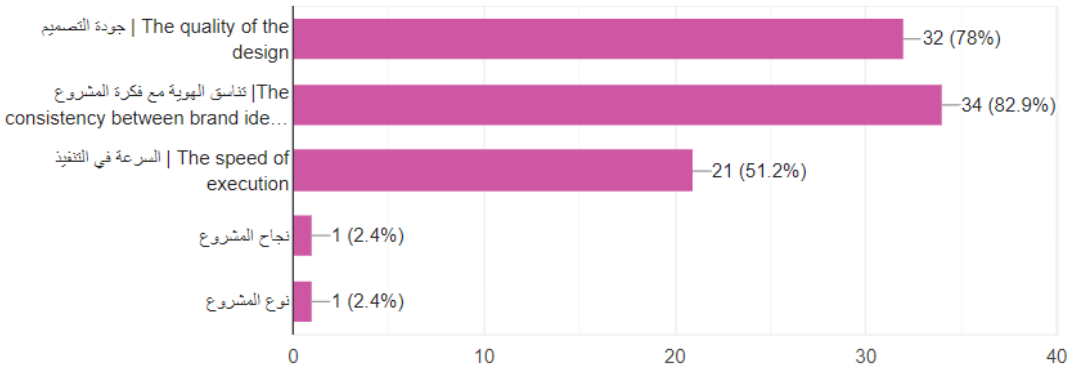


Figure A.53: Question 16 Results

General Information | معلومات عامة

نسخ

Do you think that AI-powered marketing can help achieve the goals of the 2030 vision by solving many of the challenges faced by business owners?

هل تعتقد أن التسويق المدعوم بالذكاء الاصطناعي يمكن أن يساعد في تحقيق أهداف رؤية 2030 من خلال حل العديد من التحديات التي يواجهها أصحاب الأعمال؟

136 ردًا

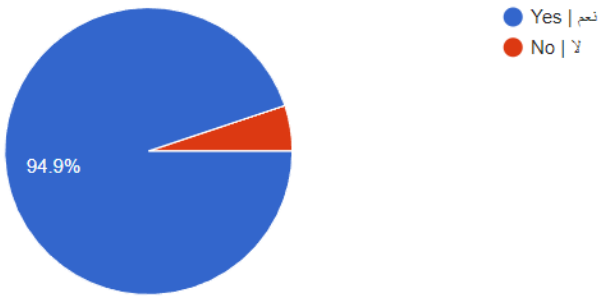


Figure A.54: Question 17 Results

If I found a tool for creating marketing content for my store, this will be a suitable option for me to rely on during marketing campaigns.

إن وجدت أداة لصنع المحتوى التسويقي لمنشأتي/ متجري، سيعد ذلك خيارا مناسبا لي للاعتماد عليه خلال الحملات التسويقية

136 ردًا

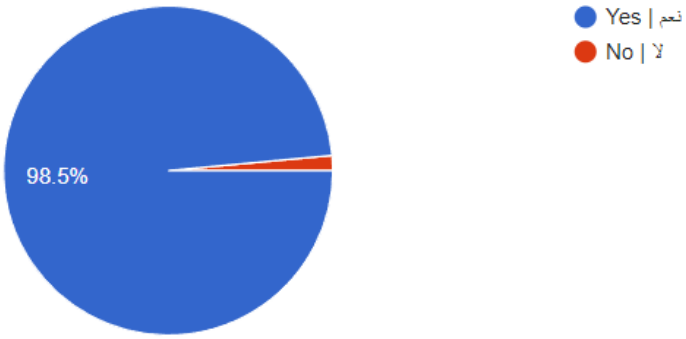


Figure A.55: Question 18 Results

?Do you have any comments, suggestions, or concerns regarding our project idea

هل لديك أية تعليقات أو اقتراحات تود مشاركتها بخصوص فكرة مشروعنا؟

35 ردًا

لا

لا يوجد

لا يعطيكم العافية □

لا التوفيق لكم

لايوجد

لايوجد

No

لا

استمرو فأنتم مبدعون

Figure A.56: Question 19 Results

اريد اعلاني و عروضي تكون متوفره في كل مكان و زمان

حبيتت فكره المشروع

المحتوى التسويقي الذي يتم توليديه يجب ان يكون فريد من عميل لاخر

مشروع جميل جداً اتمنى يخرج بأفضل من ما تتخيلون ♥♥

مشروع جميل و هادف ويمثل الرؤية وفكره جميلة واستمروا في هذا الابداع

شكرًا

التفاعل مع الجمهور اصعب شي و ان تكون طول الوقت متواجد

إضافة امكانية تعزز رغبة مختلف الأشخاص بقتراح أفكار تنمي التطوير و الإلهام لأصحاب المشروع.

No thank you

Figure A.57: Question 19 Results

بالاضافه إلى صناعة الهوية نحتاج الذكاء الاصطناعي في التسويق لعمل الخطة التسويقية مثل أن ينشيء خطة زمنية لنشر بوستات وفيديو في السوشل ميديا وغيره والتذكير بموعد تنفيذها. واقتراح أحدث طرق التسويق

فكره المشروع رائعه اتمنى انها تكون بأسعار معقوله لأن اسعار المسوّقين مرتفعه جداً

فكره جدا ممتازه الذكاء الاصطناعي الناس صارت تستخدمه في اغلب امورها فاتوقع بيكون رااائع جدا لو استخدم كاداه للتسويق مثلا يعطيني افكار، يصنع لي محتوى، او مثلا حتى يسوي لي جدول عشان احل مشكلة الاستمرارية واصير اتبع هذا الجدول وفي نفس الوقت ما اهمل دراستي...الخ

Figure A.58: Question 19 Results

نسخ

If you have any questions or concern, please provide your contact number or email:
إذا كان لديك أي أسئلة أو استفسارات، يرجى تقديم رقم هاتفك أو بريدك الإلكتروني:

16 ردًا

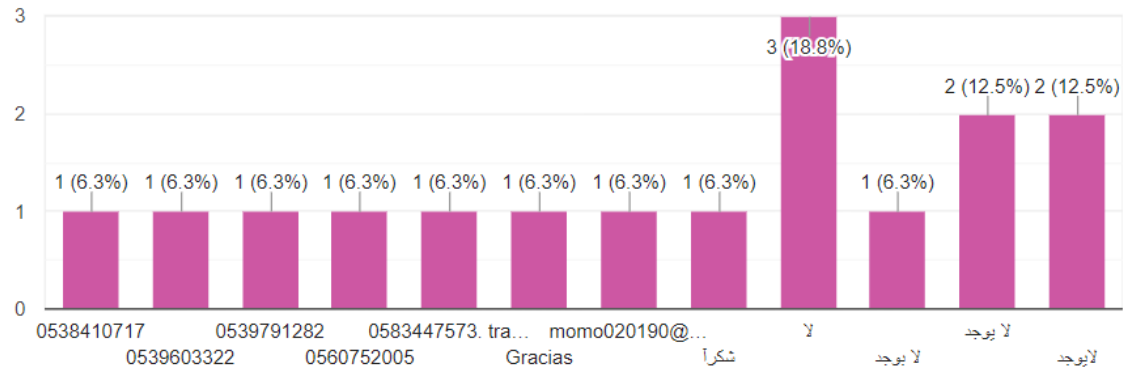


Figure A.59: Question 20 Results

Appendix B: User Guide

Welcome to the installation guide for Mosaweeq. This guide will walk you through the steps to install and run the project on your device. By following these instructions, you'll be able to access our website and utilize its features efficiently. Let's get started!

B.1 Installation and Running of Mosaweeq

To install and run the project on your device, follow these steps:

B.1.1 Prerequisites

1. Python: Ensure you have Python installed on your system. You can download it from python.org.
2. Virtual Environment: It's recommended to use a virtual environment to manage dependencies. You can create one using venv.
3. Visual Studio: Ensure you have Visual Studio installed, or use any preferred IDE or text editor.

B.1.2 Installation Guide

- Download the Zipped Folder
    - Obtain the zipped folder containing the project files.
    - Extract the zipped folder to your desired location on your device.
- Set Up the Virtual Environment
    - Open a terminal in visual studio code.
    - Navigate to the project directory.
    - Create a virtual environment by running: python -m venv venv
    - Activate the virtual environment
        - On windows: venv\Scripts\activate
        - On macOS/Linux: source venv/bin/activate
- Install Required Dependencies
    - Ensure you are in the project directory and the virtual environment is activated.
    - Install the required Python packages by running: pip install python
- Configure the SQLite Database
    - The project uses an SQLite database. you can install its extensions from visual studio code. Ensure the database file (Mosaweeq.db) is in the project directory. If not, you may need to run the “createDatabase.py” file to create the database.

- Run the Flask Application
    - Set the Flask app environment variable:
        - On windows: set FLASK_APP=app.py
        - On macOS/Linux: export FLASK_APP=app.py
    - Run the Flask development server: flask run
    - The server should start, and you'll see output indicating the application is running on http://127.0.0.1:5000/.
- Access the Website
    - Open a web browser and navigate to http://127.0.0.1:5000/ to access the website.

B.2 Utilizing Mosaweeq

After Accessing the website you will be able to view the homepage of Mosaweeq, which displays overview of website features, about us, our service and contact information. The following figures shows the homepage of Mosaweeq:

Figure B0.1: Home page

From the first figure, in the header you will be able to view a signup and sign in buttons that allow you to access our features. And this is the footer in the homepage:

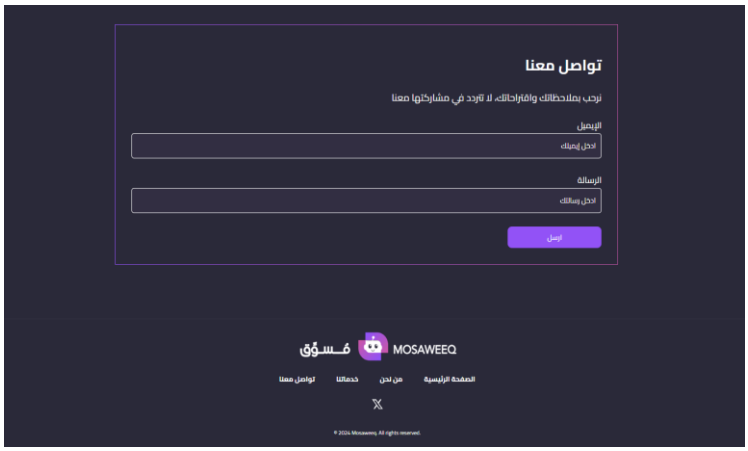


Figure 0.2: footer

After signing in you can play the following features:

1. Create Brand Identity

   You can access this feature by clicking on "أنشئ هويتك التجارية", then the following page will be displayed.

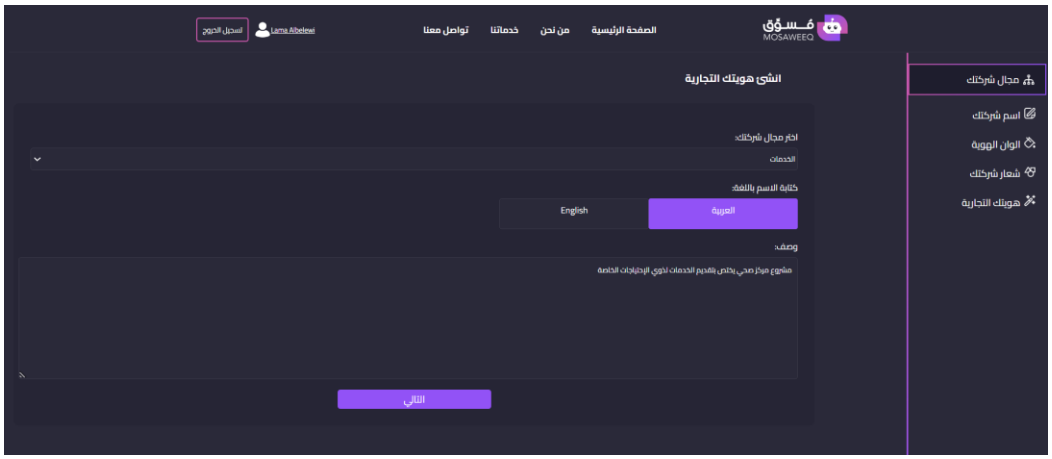


Figure 0.3: brand identity page 1

- Step 1: Business Category and Description
  - Choose the category of your business from the provided options.
  - Select your preferred language.
  - Enter a short description of your business.
  - Click "Next" to proceed.

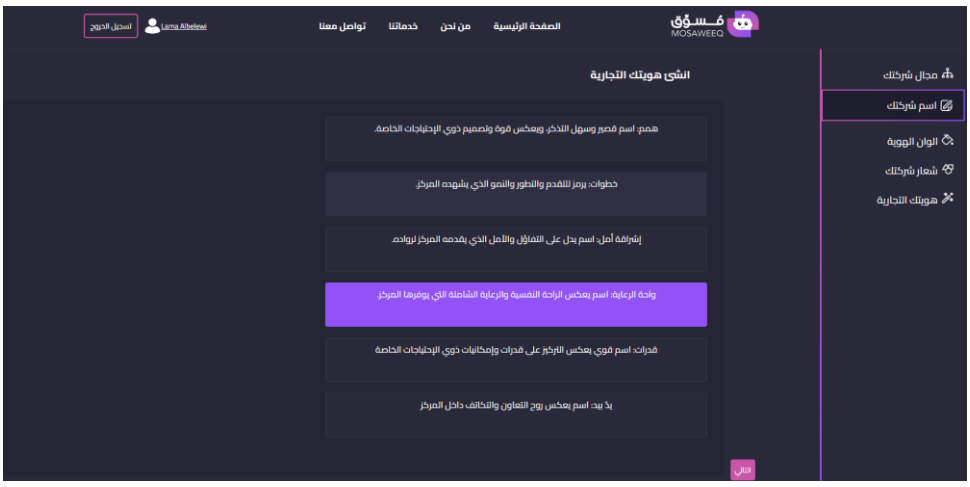


Figure 0.4: brand identity page 2

- Step 2: Generate Brand Names
  - The AI will generate a list of potential brand names based on your input.
  - Select a brand name from the list.
  - Click "Next" to continue.

- Step 3: Choose Preferred Colors
  - Select the preferred color scheme for your brand identity.
  - Click "Next" to proceed.

Figure B.5: brand identity page 3

- Step 4: Generate Suggested Logos
  - The AI will generate suggested logos based on your selected brand name and color scheme.
  - Choose your preferred logo.
  - Click "Next" to view the final brand identity.

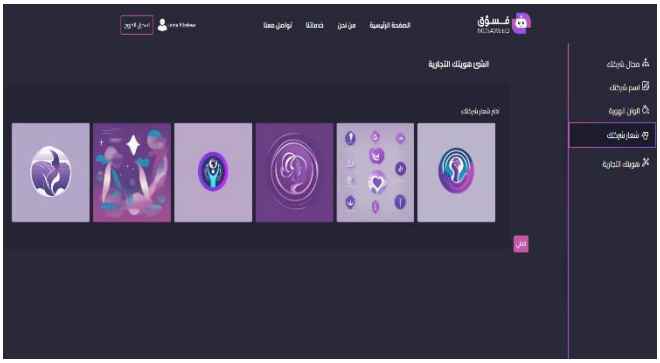


Figure 0.6: brand identity page 4

Figure 0.7: brand identity page 5

- Step 5: View Final Brand Identity
  - Review the complete brand identity including the selected name, color scheme, and logo.

2. Generate Marketing Content

You can access this feature by clicking on "أنشئ محتوى تسويقي جديد", then the following page will be displayed.

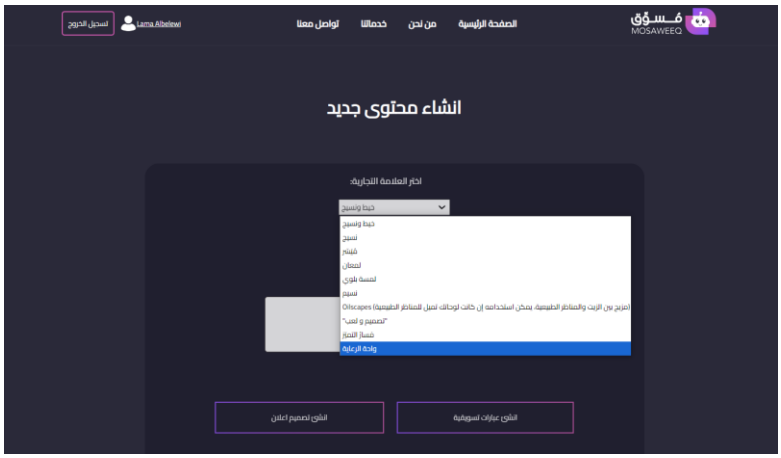


Figure 0.8: generate marketing content page

- Select the brand you want to generate the marketing content for.
- Write a short description of the marketing content you need.

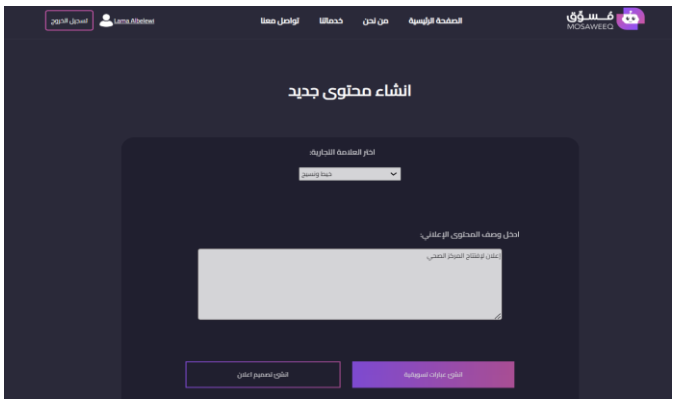


Figure 0.9: generate marketing content page

- Text Marketing Content
  - Select the option to generate text marketing content.
  - The AI will produce a marketing text suitable for social media posts.

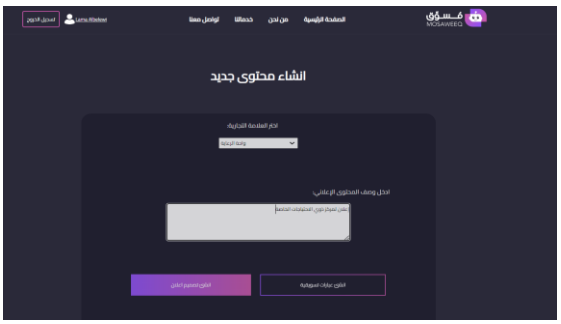


Figure 0.10: generate marketing content page

- Image Marketing Content
  - Select the option to generate an image that includes your product, logo, and marketing text.
  - The AI will create an image ready for social media posting.

You can click on save to either copy the marketing tagline to paste it online or save the post to your device.

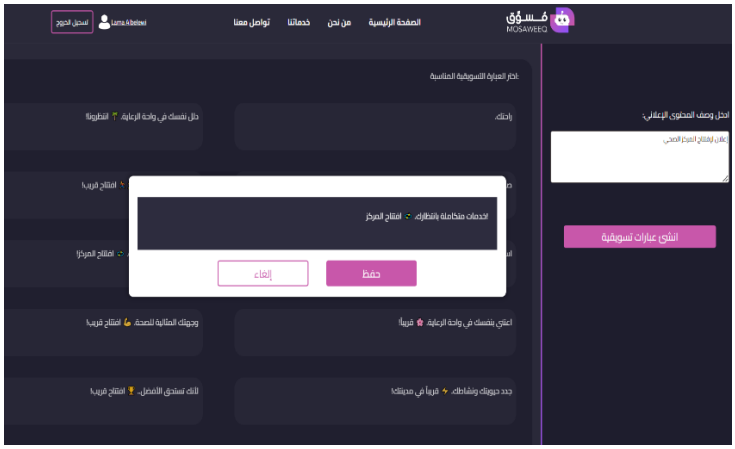


Figure 0.11: marketing tagline

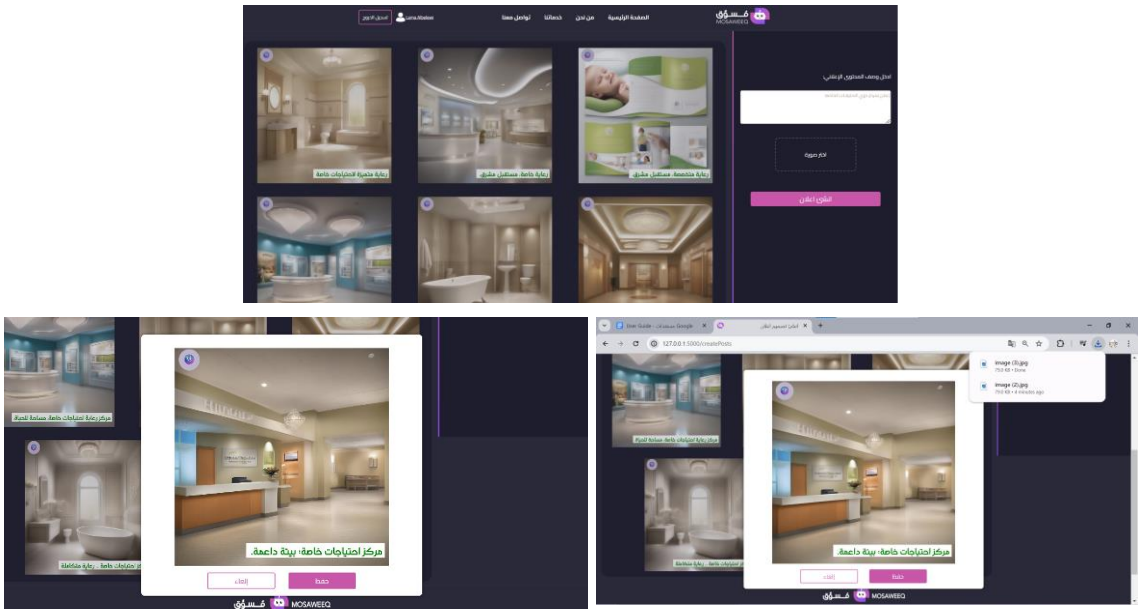


Figure 0.12: marketing image

3. Generate Marketing Content Based on Events

You can access this feature by clicking on "أنشئ محتوى لمناسبة قادمة", then the following page will be displayed.

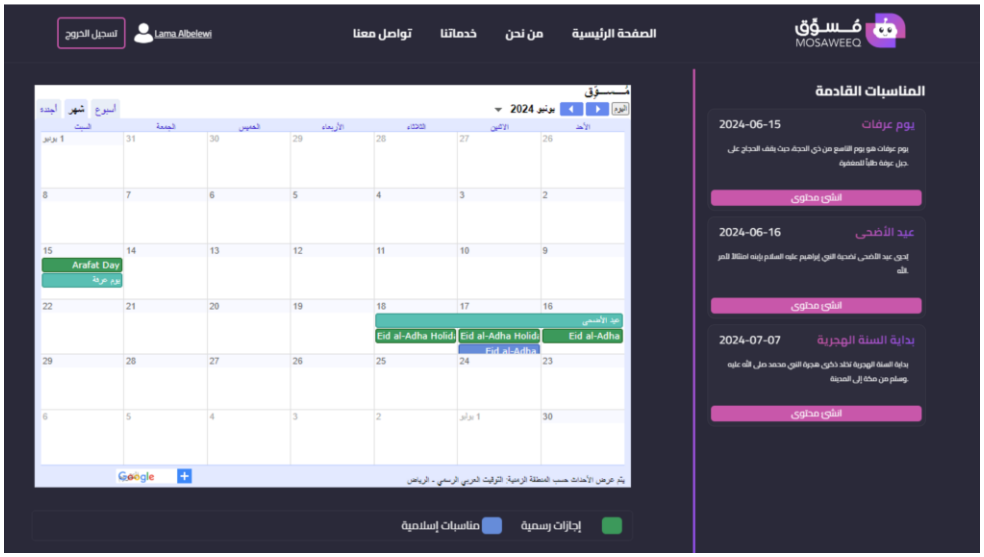


Figure 0.13: Generate Content for an Upcoming Event Page

- Step 1: View Events Calendar
  - Browse the calendar displaying upcoming events in Saudi Arabia along with their descriptions.
- Step 2: Select an Event
  - Choose an event from the calendar.
- Step 3: Generate Event-Based Marketing Content
  - Click "انشئ محتوى" to generate marketing content tailored to the selected event.
  - Then, the following procedures will be the same as mentioned in creating marketing content above:
    - Click the brand you want to make the marketing content about.
    - Add a short description for the advertisement post you want.
    - choose whether you want marketing images to post or text.
    - The AI will create appropriate marketing material for the event.

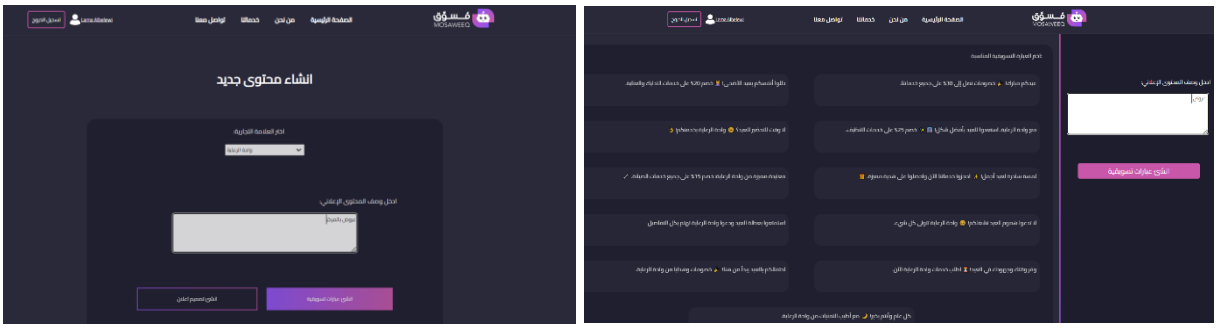


Figure B.14: Generate Event-Based Marketing Content

Additional Pages

- Profile Page: Displays user information.
- Sign Out: Button to sign out from the website.

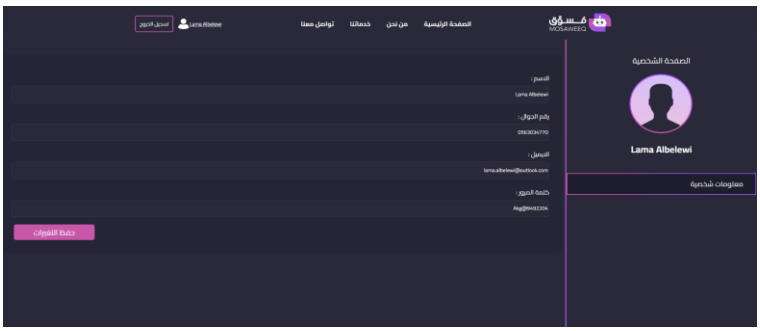


Figure B.15: profile page

This guide provides a comprehensive overview to help you efficiently utilize our website's features. If you need any further assistance, feel free to reach out to us through the contact information provided on the Home Page.