DESIGN AND SIMULATION OF A PHISHING PORTAL WITH GOOGLE WORKSPACE

BY

OGHEDEGBE ODION DANIEL

PSC2105371

DEPARTMENT OF COMPUTER SCIENCE,

FACULTY OF PHYSICAL SCIENCES,

UNIVERSITY OF BENIN,

BENIN CITY,

EDO STATE, NIGERIA.

NOVEMBER 2025

DESIGN AND SIMULATION OF A PHISHING PORTAL WITH GOOGLE WORKSPACE

BY

OGHEDEGBE ODION DANIEL

PSC2105371

A PROJECT REPORT SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE, FACULTY OF PHYSICAL SCIENCES, UNIVERSITY OF BENIN, BENIN CITY

IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE AWARD OF A BACHELOR OF SCIENCE (B.Sc.) DEGREE IN COMPUTER SCIENCE

NOVEMBER 2025

CERTIFICATION

This is to certify that this project work was carried out by **OGHEDEGBE ODION DANIEL** with Matriculation Number **PSC2105371** under my supervision. It is adequate and satisfactory, both in scope and content, for the award of Bachelor of Science (B.sc) Degree in Computer Science of the University of Benin

---

DR. E. NWELIH
Project Supervisor

DATE

APPROVAL

This project work is hereby approved in partial fulfilment of the requirements for the award of Bachelor of Science (B.Sc.) Degree in Computer Science from the University of Benin.

DR. (MRS.) A. ROSEMARY USIOBAIFO
Head of Department

DATE

DEDICATION

This project is dedicated to God Almighty for giving me the strength and wisdom to see it through to completion, and even throughout my stay in the University of Benin (UNIBEN). It is also dedicated to my parents; Mr and Mrs Oghedegbe and my Aunties and Uncles ; for their love, support and guidance throughout my academic journey.

ACKNOWLEDGEMENT

My utmost acknowledgement goes to God Almighty for giving me the grace, wisdom and direction throughout my journey academically. I would like to express my gratitude to my project supervisor, Dr. E . Nwelih for his consistent guidance towards ensuring the successful completion of this project.

I would also like to specially thank my project coordinator Dr. (Mr.) Maxwell Osagie, and other lecturers in the Department of Computer Science whom have impacted me all through my years in this institution.

I also want to appreciate those who contributed to the success of this project: Omoikhoje Macaulay De-Great, Akingbile Gabriel Ayooluwabami, Akpowenre Ogaga Daniel, and Victor Chibundu Ezeani. I would also like to thank my family and friends for their support and consistent guidance throughout this project.

ABSTRACT

Traditional cybersecurity awareness programs often struggle with engagement, scalability, and the ability to measure real behavioral change among users. To address these challenges, this project presents the development of a Phishing Simulation and Awareness Portal using Google Workspace tools. The system is designed to simulate real-world phishing attacks in a controlled environment and educate users through interactive, micro-based lessons. Built with Google Apps Script, Gmail, and Google Sheets, the platform automates the creation, deployment, and tracking of phishing campaigns, while maintaining real-time event logging and analytics. It also provides administrators with a simple, centralized interface for managing email templates, campaign configurations, recipient data, and awareness metrics. By integrating simulation and training within the familiar Google ecosystem, the system eliminates the need for external applications and ensures ease of deployment across organizations. Ultimately, this project aims to enhance user vigilance, reduce susceptibility to phishing threats, and promote a culture of proactive cybersecurity awareness through continuous education and data-driven feedback.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER ONE

INTRODUCTION

1.0 Background Study

The increasing prevalence of phishing attacks poses a significant threat to individuals, organizations, and institutions across the digital landscape. Phishing remains one of the most common methods used by cybercriminals to manipulate users into divulging sensitive information, such as login credentials, credit card details, and personal data. According to Symantec (2023), over 90% of security breaches begin with a phishing email, highlighting the urgent need for effective awareness and prevention strategies. Despite advancements in email security and spam filtering technologies, many users continue to fall victim to well-crafted phishing campaigns that exploit human psychology rather than technical vulnerabilities.

This study focuses on the development of a Phishing Simulation and Awareness Portal using Google Workspace Tools, designed to evaluate users' susceptibility to phishing attempts and provide targeted educational feedback. The system simulates realistic phishing scenarios, tracks user interactions, and delivers micro-learning lessons to reinforce cybersecurity awareness by leveraging Google Apps Script, Gmail API, and Google Sheets,. As noted by Mittal et al. (2022), integrating simulation-based training into organizational workflows enhances user alertness and builds a culture of security-conscious behavior. The project thus aims to combine automation, data-driven insights, and accessible web technologies to support proactive phishing education and reduce human-related security risks.

The platform will consist of two major components:

- **Administrator Side (Cyber/IT Staff):** Allows simulation of phishing emails, tracks clicks and responses, and generates reports.
- **User Side (Employees or Staff):** Receives simulated phishing emails and automatically receives awareness messages or short video tips when they fail the test

1.1 Motivation

The increasing prevalence of phishing attacks targeting individuals and organizations has highlighted the urgent necessity for proactive cybersecurity awareness frameworks that transcend traditional defense mechanisms. Phishing continues to be one of the most pervasive and effective social engineering tactics employed by cybercriminals to manipulate users into divulging sensitive information or granting unauthorized access to secured systems. According to the Verizon Data Breach Investigations Report (2023), phishing accounts for over 36% of all recorded data breaches, thereby emphasizing that technological safeguards alone are inadequate in mitigating this persistent threat. The most exploited component in the cybersecurity chain remains the human factor,particularly users' awareness levels, judgment, and behavioral tendencies when interacting with digital communication systems.

Despite growing awareness of phishing threats, many organizations continue to depend on static, theory-based awareness programs such as routine workshops, generic email advisories, and infrequent seminars. However, these traditional methods have proven insufficient in producing lasting behavioral change. Kaur and Singh (2021) observed that individuals who receive only theoretical training often fail to recognize deceptive phishing messages when exposed to real-life scenarios. This underscores the limitations of conventional educational approaches that lack interactivity, contextual learning, and continuous engagement. As a result, there is a pressing demand for dynamic systems that

can reinforce user awareness through practical simulations, adaptive feedback, and measurable performance analytics.

The high cost and technical complexity associated with most commercial phishing simulation tools present significant barriers for small and medium-sized enterprises, educational institutions, and non-profit organizations. These sectors, despite their vulnerability, often lack the financial and infrastructural resources to implement advanced cybersecurity awareness platforms. The utilization of Google Workspace tools such as Google Apps Script and Google Sheets offers a cost-effective and scalable alternative for developing a functional phishing simulation system. This approach leverages widely used and user-friendly tools, thereby reducing implementation challenges while promoting inclusivity and accessibility across varying organizational levels.

The motivation behind this study is the aspiration to design and develop an affordable, interactive, and sustainable Phishing Simulation and Awareness Portal that fosters continuous cybersecurity education through experiential learning. the project aims to bridge the gap between theoretical knowledge and practical response behavior by integrating real-time simulations, user performance tracking, and micro-learning modules within a single framework. The system is envisioned to strengthen organizational resilience, cultivate informed digital practices, and establish a culture of vigilance where end users become proactive defenders against phishing and other forms of cyber deception.

1.2 Statement of Problem

Phishing has emerged as a leading and harmful type of cybercrime, exploiting human psychology instead of technical flaws to breach systems and acquire sensitive data. Cybercriminals are adopting increasingly advanced strategies like domain spoofing, fake

websites, and tailored emails to trick users into revealing confidential information or taking harmful actions. According to the Anti-Phishing Working Group (APWG) 2023 report, phishing incidents are growing significantly, with millions of new phishing sites detected every quarter. This trend indicates that traditional security measures such as spam filters and antivirus software are ineffective when users lack understanding of online deception.

Standard cybersecurity awareness programs often fail to tackle this problem, as they usually consist of sporadic sessions or static content that focus more on information dissemination than on fostering behavioral changes. Users end up lacking the hands-on knowledge required to recognize and respond effectively to phishing attempts. Research by Kaur and Singh (2021) highlights that individuals receiving theoretical instruction frequently fall victim to phishing simulations shortly after training, illustrating a gap between awareness and practical application. This lack of ongoing, interactive education leaves organizations vulnerable to financial, reputational, and operational threats.

Existing commercial phishing simulation tools are either prohibitively expensive or complicated, making them unsuitable for small and medium-sized businesses, educational institutions, and non-profits which have limited budgets and IT resources which leaves these entities with a lack of structured cybersecurity awareness initiatives. There is, therefore, a pressing need for a cost-efficient, user-friendly phishing simulation solution that can be easily integrated into organizations through accessible digital platforms.

This project proposes to build a Phishing Simulation and Awareness Portal using Google Workspace tools, notably Google Apps Script, Gmail, and Google Sheets. The envisioned system will create realistic phishing scenarios to evaluate and improve users' awareness levels while tracking responses and providing customized micro-learning content based on performance. The aim of this portal to foster continuous behavioral enhancement and

strengthen organizational defenses against phishing threats by bridging the gap between conventional training and real-life experiences. This study intends to provide an affordable, data-driven, and sustainable strategy to elevate cybersecurity awareness across various user groups and institutions.

1.3 Aims & Objectives

The aim of this project is to develop and simulate a phishing portal and awareness system using Google Workspace tools.

The objective is to improve user awareness and reduce vulnerability to phishing attacks in small-scale organizational environments.

1.4 Scope of Research

This project focuses on the implementation of a phishing simulation and awareness system, built to educate users on how to recognize and respond to phishing threats, using small organizations that rely on Google Workspace tools.

1.5 Research Methodology

The Object-Oriented Analysis and Design Methodology (OOAD) was employed in this project to methodically analyze and design the system as an assemblage of interconnected objects and components. The project utilized an observational approach for data collection, scrutinizing documentation related to Google Workspace technologies while performing a comprehensive examination and evaluation of phishing simulation tools such as Cofense and KnowBe4.

This structured methodology facilitated the clear delineation of user roles, system components, and their interactions within the platform. It acted as a framework for the development of a prototype system capable of simulating phishing emails, monitoring user responses, and delivering automated feedback to enhance awareness regarding phishing

threats. The design was aimed at producing a streamlined and user-friendly system, and the analytical phase played a crucial role in uncovering existing challenges related to cost and accessibility faced by small organizations.

1.6 Research Significance

The importance of this research lies in its ability to enhance cybersecurity awareness and resilience through a novel, practical, and economically viable educational method. Phishing remains a significant threat to information integrity, with human mistake recognized as the critical vulnerability in numerous security incidents. This study proposes the development of a Phishing Simulation and Awareness Portal utilizing Google Workspace tools, presenting a sustainable remedy that connects theoretical cybersecurity training with practical behavioral responses.

In contrast to traditional awareness initiatives that depend on isolated training sessions or passive educational materials, this project prioritizes experiential learning,affording users the chance to engage in realistic phishing simulations, identify warning indicators, and gain insights through immediate feedback. Such a methodology encourages ongoing education and solidifies long-term changes in behavior, which can therefore diminish vulnerability to phishing schemes.

Moreover, the deployment of this system by utilizing Google Apps Script, Gmail, and Google Sheets guarantees ease of access, scalability, and cost-effectiveness. These tools are widely accessible within educational establishments, small enterprises, and non-profit organizations, facilitating straightforward implementation without the necessity for advanced technical skills or extra infrastructure. Consequently, the project allows under-resourced entities to effectively enhance their cybersecurity defense mechanisms.

Within the academic sphere, this investigation contributes to the advancing domain of cybersecurity education and simulation-based learning by illustrating how open and cloud-based technologies can be repurposed for awareness-building efforts. Practically, it aids organizations in cultivating a proactive ethos of vigilance,transforming users into the primary line of defense against cyber threats.This research in the long run offers both a technical and educational framework to bolster human-centered security practices within today's digital environment.

CHAPTER TWO

LITERATURE REVIEW

2.0 Introduction to the Literature Review on Phishing Attacks and Related Concepts

The quick growth of digital systems and widespread use of cloud computing have increased the risk of cyber attacks for individuals and companies. Phishing is a common and tricky threat. These attacks use tricks and flaws in systems to fool people into giving up important info, like passwords or money. This often happens through fake messages that look like they're from trusted sources.

This section looks closely at what experts and professionals have written about phishing attacks, how to spot them, ways to prevent them, and real-world examples. It focuses on using Google Workspace to create a sample phishing gateway and also clarifies how current knowledge can guide the creation of an ethical tool for education and training.

The review of the literature starts with how phishing has changed over time, listing its types and methods, checking detection systems, and rating defenses. Later parts explore how the mind works, social and economic effects, case studies, legal structures, and new trends influenced by machine learning. Extra attention is given to phishing in cloud environments, keeping in mind the project's focus on Google Workspace. This collection includes tools like Gmail, Forms, and Sites that can be used for harmful acts or protective steps. This section points out gaps in research, like a lack of standard simulation systems that combine machine learning with user training by examining literature reviews and studies.It also suggests how the portal to be developed can fix these problems and promises careful methods and relevance to current discussions about cybersecurity.

2.1 The Historical Evolution and Chronological Development of Phishing Attacks

Phishing attacks first appeared in the mid-1990s, with early online communities on services like AOL being the initial victims. Attackers used simple scripts to steal user passwords, a method known as AOHell. Early texts on cybersecurity note that these simple methods progressed into more elaborate and carefully planned attacks by the early 2000s. The growth of electronic commerce and financial services at this time gave attackers a more direct financial motive. By 2004, the Anti-Phishing Working Group (APWG) was formed to monitor the growing number of phishing attempts. Their reports showed a rapid increase in deceptive emails, growing from a few thousand to millions, coinciding with the internet reaching over a billion users.

Alkhalil et al. (2021) offer a detailed breakdown of how phishing works, separating it into distinct stages: identifying potential targets, sending out deceptive messages, obtaining sensitive information, and converting that information into financial gain. They point out that attack methods shifted from wide-net scams to more focused spear-phishing attacks. This change was supported by easily accessible technology, such as phishing kits found on underground online marketplaces, that allowed even inexperienced individuals to start attacks. Naqvi et al. (2023) add to this understanding by reviewing 248 papers published between 2018 and 2023. They attribute the rise in attacks to automated tools and the use of new platforms such as mobile applications and social networks. The COVID-19 pandemic further accelerated this trend. The shift to remote work led to cloud services becoming attractive targets for attackers. Butt et al. (2022) documented an increase in cloud-centered phishing attacks after 2020, with attackers targeting these platforms to steal identities as more people started depending on virtual tools for work and communication.

The recent APWG Phishing Activity Trends Reports for 2024-2025 indicate a large increase of 4,151% in misleading emails since the spread of generative AI in 2023. The first quarter of

2025 saw more than one million attacks, marking the highest level since late 2023. This jump is also reflected in Verizon's 2025 Data Breach Investigations Report (DBIR), which examined 22,052 security incidents. The report found that 36% of breaches involved phishing techniques, including business email compromise (BEC), which results in over $2.9 billion in losses annually. These shifts show phishing's ability to adapt, changing as security steps like multi-factor authentication (MFA) become more common. Attackers now bypass MFA by using adversary-in-the-middle (AitM) tactics. Understanding these changes is essential for setting up a Google Workspace portal that simulates past and present attack methods to develop better protective measures.

2.2 Comprehensive Classification of Types and Operational Techniques Employed in Phishing Attacks

Phishing attacks use many tricky ways to fool people by finding weak spots in how they think, act, and use technology. Sorting phishing attacks into different types helps us to learn how they work and make better defenses. This part breaks down the main kinds of phishing, explains how they function, how they've changed over time, and compares them based on what different studies. It uses information from reviews like those by Alkhalil et al. (2021) and Naqvi et al. (2023), which look at over 300 research papers. The discussion looks at how attackers plan their moves and why people fall for them, and also points out the flaws in current systems for sorting attacks when dealing with new kinds that mix old methods with artificial intelligence. A recent study in 2025 by Catal et al. (2025) on AI-improved phishing shows that the lines between attack types are getting fuzzy, which means we need to change how we classify them for future study.

The methods used in phishing usually depend on manipulating people using social engineering. This means attackers take advantage of quick thinking to avoid logical analysis of the situation. Research from tests in real-world work environments shows that these attacks succeed between 5% and 40% of the time. This depends on things like how personalized the message is and when it's sent. Looking closely at these methods proves that we need to use ideas from psychology, computer science, and criminology to really understand why they are so powerful.

2.2.1 Detailed Examination of Email-Based Phishing as the Predominant Vector in Cyber Deceptions

Email phishing is still the most common and cheapest way for criminals to do cybercrime. Data from the Anti-Phishing Working Group (APWG) and Verizon's Data Breach Investigations Report (DBIR) shows that email phishing makes up about 80-90% of all reported phishing incidents.

Thakur et al (2023) say there are many ways to do it, from changing the sender's information using fake SMTP headers to adding harmful things like attachments with ransomware or links that take you to fake websites to steal your login information. Alkhalil et al (2021) agree, saying that it happens in stages, starting with finding email addresses from data breaches and then tricking people into clicking links by making them feel like they need to act fast. They tested this by looking at real email data, like the Enron corpus, and using it to simulate phishing attacks.

New artificial intelligence (AI) has changed email phishing a lot. Criminals can now make very personalized emails that look like real messages with good grammar and relevant information. Hoxhunt's 2025 Phishing Trends Report looked at over 386,000 emails and

found that AI-made emails are only about 0.7-4.7% of all attacks, but they are 23% more successful when used to target executives. These emails often use information from social media to make them seem more personal. Butt and others (2022) also looked at emails that use cloud services and used machine learning models like support vector machines (SVM) to find them with 99.62% accuracy. They looked at things like the language used (how positive or negative it is) and if the links seemed strange (like if the URL didn't match the text). One problem with these studies is that they mostly use English emails, so they might not be as good at finding phishing emails in other languages.

It's important to know that email phishing works because it plays on people's psychology. For example, criminals might pretend to be someone important (like the CEO in a business email compromise) or make you feel like you need to act fast (like a limited-time offer). Studies show that older people are more likely to fall for these tricks. A 2024 study that looked at 50 other studies found that personality traits can make people more vulnerable to phishing. For the portal, you can simulate these emails using Google Workspace's Gmail API. This lets users get real-time feedback on how well they can spot phishing emails. It also helps to fill gaps in the current research by using AI-made emails that change based on the user.

2.2.2 Detailed Examination of Email-Based Phishing as the Predominant Vector in Cyber Deceptions

Email is still the main way social engineering attacks get in because lots of people use it, businesses depend on it, and it's easy to fake trustworthy signs. Data from the industry shows that many incidents start when someone clicks a harmful link or opens an infected file they got in an email (Verizon DBIR 2024). From a study carried out by Proofpoint in 2024,it says

that attackers keep trying to steal credentials using fake branded landing pages and OAuth consent scams.

Technically, attackers make sure their emails get delivered by misusing real infrastructure. For example, they might send emails from accounts they've hacked, use cloud storage links, or hide links in PDFs. They also try to make their emails more convincing by copying internal style guides, ticket numbers, and workflows specific to certain roles.

Business Email Compromise (BEC) campaigns make things even worse by using fake reasons for wire transfers and tampering with supplier invoices. Often, these attacks don't involve any malware, so they can get past standard antivirus and EDR protections (Palo Alto Networks BEC). The main reason email is so popular for attacks is that it's so common, people trust it, and it's hard for defenders to spot the signs, even when secure email gateways are in place.

To really understand why email is such a big problem, it's important to look at each of these things in more detail:

- Ubiquity: Almost everyone uses email, both for work and personal stuff. This means attackers have a huge pool of potential victims. No matter how careful some people are, there are always others who might fall for a scam.

- Trust: People tend to trust emails, especially if they look like they're coming from a known source, like a coworker or a company they do business with. Attackers take advantage of this trust by faking email addresses and imitating the look and feel of legitimate emails.

- Low Signal-to-Noise: With so many emails going in and out every day, it's hard for security systems to pick out the malicious ones. Phishing emails can look very similar to legitimate emails, making it difficult to tell the difference. This is where attackers get crafty at hiding their tracks.

The combination of these three things makes email a very attractive target for attackers. They can reach a lot of people, trick them into doing what they want, and hide their activities in the noise of everyday email traffic. Even with security measures in place, email remains a weak spot for many organizations. It's a constant battle to stay ahead of attackers who are always coming up with new ways to exploit this channel.

2.2.3 Exploration of Cloud-Based Phishing Attacks Targeting Platforms Like Google Workspace and Similar Ecosystems

The increasing popularity of cloud collaboration tools, like Google Workspace, Microsoft 365, and Slack, has drastically changed how organizations operate. These platforms combine things such as email, file storage, document sharing, messaging, and scheduling into one digital space.

While this shift has improved efficiency and made remote work easier, it has also opened up more opportunities for phishing attacks. Phishing, which used to mainly target email, now uses multiple methods within these cloud platforms. Attackers take advantage of the trust people place in shared links and documents.

To elaborate on the changes in organizational workflows, the integration of these platforms has led to a more fluid exchange of information. Teams can now work together on documents in real-time, regardless of their physical location. Communication is quicker through instant messaging, and scheduling meetings is more convenient with integrated calendars. This interconnectedness has reduced the reliance on traditional methods of communication and project management.

The broadened attack surface is a serious concern. Attackers are now targeting various entry points within these cloud platforms, not just email. For example, they might send phishing

links through direct messages on Slack or embed malicious code in shared documents on Google Drive. The assumption that internal communications and shared files are safe makes users more susceptible to these attacks.

The move to cloud-based collaboration has brought lots of advantages, but it also requires a heightened focus on security awareness and training. Organizations need to educate their employees about the various forms that phishing attacks can take within these platforms and the steps they can take to protect themselves. This includes verifying the authenticity of shared links and documents, being cautious of suspicious messages, and keeping software up to date with the latest security patches.

2.3 Cognitive Biases and Decision Making Under Uncertainty

Phishing works because people often make quick judgments with limited focus under pressure, not because they lack intelligence. Attackers today craft lures that take advantage of well-known mental shortcuts, or heuristics, leading people to act before thinking. Current cyber-research shows that heavy workloads, like overflowing inboxes or tight deadlines, cause people to rely on these heuristics instead of careful reasoning; the ideal situation for phishing to succeed. Based on dual-process and heuristic-systematic theories, recent research connects biases like deference to authority, a sense of urgency, scarcity, and familiarity to increased rates of clicks and submitting credentials. This has direct implications for designing training programs and setting simulation schedules.

2.3.1 Key biases that attackers commonly exploit

Authority and hierarchy: Messages that seem to come from bosses, financial departments, HR, or IT use the authority bias. Studies show people are more likely to comply and less likely to question requests when they appear to come from someone in charge, such as fake

payment approvals from a CFO or security updates from IT. This is especially true when time is short.

Urgency and time pressure: Subject lines like “ACTION NEEDED: Account disabled in 24 hours” force people to make rapid decisions, skipping steps like checking links or headers. Research shows that urgency reduces suspicion and increases the likelihood of clicking by preventing careful thought.

Scarcity and fear of loss: Messages like “Only 2 seats left in mandatory training” or “final payroll confirmation” exploit the fear of missing out. These prompt workers to act quickly, particularly if the action seems routine, like approving, confirming, or downloading.

Familiarity and social proof: Impersonating well-known brands like Google or Microsoft, or colleagues with messages like “see attached, as per your request,” builds trust. Studies show that trust by association, especially from compromised internal accounts, greatly increases success rates.

2.3.1.2 Decision modes: Why context matters more than knowledge.

Recent studies show that how people process information at the moment is a better predictor of detection accuracy than what they know in theory. Findings show that using heuristics, or quick, cue-driven thinking, leads to poorer detection of phishing attempts. Mindful, systematic thinking improves accuracy, and emotional state also influences results. In short, employees who are rushed, distracted, or emotionally charged are more likely to miss warning signs, even after training.

Notably, an experiment found that outcomes depend on how tasks are framed and the situation, showing that solutions should shape situations by reducing pressure, adding steps, and making cues visible, not just telling people to “think harder.”

2.3.1.3 Attention, fatigue, and the gap between knowing and doing:

Surveys show that many people know the rules but still take risks for convenience, speed, or to meet perceived demands. Reports show the majority of workers admitted to risky actions despite knowing better, often citing urgency and convenience. Security fatigue, caused by constant warnings and generic training, further normalizes quick, uncritical clicks.

Related studies support this, noting that distraction, pressure to act fast, and burnout are cited more often than sophisticated attacks as the main causes of breaches. This suggests that understanding attacker psychology is more important than just technical defenses in preventing everyday breaches.

2.3.1.4 Design implications for this project:

. **Bias-aware templates**: Vary authority, urgency, scarcity, and familiarity cues in campaigns and measure how different departments respond. This turns theory into practical risk profiles.

Add friction: Include just-in-time warnings and visual risk indicators in training emails and landing pages. Research shows these cues can improve detection without overwhelming people.

. Focus on context: Since time pressure and workload affect results, schedule simulations at different times to reflect real-world risks and teach people to pause, verify, and then act, rather than just providing facts.

. Close the loop: Offer immediate, specific feedback after a simulated click, explaining what bias was triggered and what to check next time. This reduces fatigue and promotes reflective habits.

2.3.2 Emotional Triggers and Social Influence in Phishing

While cognitive shortcuts do a lot to explain phishing, emotions are just as strong in shaping how people react. Attackers are using feelings such as fear, curiosity, excitement, or caring to get past rational thought. Really, phishing works not because it's complex, but because it can play on human feelings when someone is about to act.

(i) Fear and worry as tools

Fear is still a very good way to start a phishing attack. Statements like “Your account will be closed” or Missing payroll info—last chance trigger the instinct to avoid threats. Studies show that when fear is used with a sense of urgency, people are much less likely to be doubtful, leading to more clicks. This fits with Protection Motivation Theory (PMT), which says that when people think something is serious and they are at risk, they want to take action. Often, they don't check things out if the solution seems easy (like clicking a link to protect their account).

(ii) Curiosity and rewards

Using curiosity and rewards can also be very effective. Attackers often send messages that seem exclusive or private, playing on the normal desire to know things or get something good. Tests show that phishing that promises a reward (like a bonus, scholarship, or grant) gets more people involved than regular messages. These kinds of promises are very appealing to students, charities, and small businesses because they connect with what these groups really need.

(iii) Caring and helping

More and more studies are pointing out that triggers based on helping others—like requests to assist coworkers, confirm important demands, or support charities are being used as weapons. A report in 2024 said there were more cases of empathy phishing, where attackers pretend to

be coworkers who need help (‘ I need help urgently while traveling’) or ask for quick donations. Charities and schools are especially at risk here because they value helping others and being responsive, which makes it mentally harder to say no.

(iv) Social pressure and influence

Phishing also does well in social groups, not just with individual feelings. Messages that suggest other people have already done this or appear to come from known people use social pressure. Research says that workers are more apt to do what they're told if it seems like company policy, even if they are worried. This group behavior makes people less resistant and more likely to agree.

(v) More problems with remote work

The rise of remote work after the pandemic adds another problem: people working from home don't check things as much. They can't just ask a coworker if something seems wrong. This, along with tiredness from being online all the time, makes fear, urgency, and caring even stronger. A survey in 2023 showed that remote teams had a 30% higher rate of stolen login info due to phishing than teams working together in person. The survey said this was because there were fewer personal checks.

2.3.3 Risk Perception,Security Fatigue,and Habitual Behaviours

Phishing works because it takes advantage of how people think and feel, but the real problem is how people see risk, get tired of constant security requests, and develop habits that make them easy targets. It's important to understand these human elements to create successful and lasting security programs.

(i) How People See Risk

How people see risk is about how they guess how likely and serious a threat is. Many studies show that people don't always see phishing risks clearly. Workers usually think phishing is less risky than malware or ransomware because it looks like a normal email and doesn't seem very technical (Alkhalil et al., 2021). On the other hand, some people think they're good at spotting phishing, which makes them too confident. A report by F-Secure in 2023 found that 59% of workers thought they could always recognize a phishing email, but 36% of those same workers were tricked by test phishing campaigns in the same three months. This difference between confidence and skill is why phishing is still a big issue.

(ii) Security Fatigue and Mental Overload

Today, workers are always seeing warnings, reminders about policies, and training they have to do. This can cause security fatigue, where people stop paying attention to security because they feel stressed or like they can't do anything right. A study by NIST in 2022 showed that constant alerts and required password changes made people feel helpless, so they were more likely to ignore signs of phishing. When security becomes just background noise, phishing attacks take advantage of this lack of attention.

For charities, schools, and small businesses with few resources, security fatigue is a bigger problem because people often have many jobs. For example, a teacher might focus on preparing lessons instead of checking if an email asking for login information is real. Similarly, charity workers who are trying to meet deadlines for donors might ignore warning signs if checking them takes too much time.

(iii) Habitual Behaviors and Trust in Automation

People often depend on routines, like quickly clicking links, opening attachments, or replying to emails that look familiar, especially when they get a lot of emails. These habits make it easier for phishing attacks to work. Research based on the Dual-Process Theory of decision-

making shows that people use System 1 (automatic, fast) thinking for routine tasks instead of System 2 (analytical, slow) thinking, which would be better for spotting phishing (Naqvi, 2023).

To make things worse, people trust technology too much. They think that email filters or antivirus programs will catch everything bad, so they don't feel as responsible for checking things themselves. But phishing is often successful because it targets people's trust instead of technical weaknesses.

2.4 Phishing Simulation and Awareness Programs

Phishing simulations have become a useful way to connect what people know about phishing with how they act in the real world. By copying actual phishing attacks in a safe setting, groups can see how open they are to attacks and teach people at the moment of the simulated attack. Unlike typical awareness programs like workshops, simulations teach as part of daily online activity, which can better shape behavior in the long run.

2.4.1 Evolution of Simulation Based Training

In the early 2000s, training usually meant yearly workshops or online lessons. These were not enough to deal with the changing ways of phishing. Over the last 10 years, things have changed a lot. Simple click/no-click tests have turned into platforms powered by AI. These platforms change the phishing situations based on how employees act.

Lain et al. (2021) say that doing simulation campaigns many times lowers click-through rates a lot. This is especially true when groups keep reinforcing the lessons instead of doing it just once. More recent work (Hadnagy, 2023) says that modern phishing simulations are using current information about threats to copy real attacks. This shows users the same tricks that attackers are using now. This change from general to specific training shows that the field has grown. It now focuses on being realistic to be as helpful as possible.

2.4.2 Effectiveness of Repeated Exposure and Feedback Mechanisms

Studies consistently show that being aware of phishing depends on doing things repeatedly and getting feedback. A study by Jenkins et al. (2022) found that workers who did quarterly simulations lowered their vulnerability by 46% in the first year. This was compared to groups that only tested once a year. Feedback is just as important. Workers who clicked on phishing links but got quick, helpful advice were less likely to make the same mistakes again.

This agrees with how behavioral reinforcement theory works, where constant, helpful feedback builds good habits. What this means is that instead of punishing workers for clicking, groups can use those times as learning chances. Feedback right in the simulation email or follow-up site makes sure the learning is specific and on time.

2.4.3 Ethical Considerations and Psychological Impact of Simulations

Even though they are helpful, phishing simulations bring up moral questions. Very intense campaigns, like those copying payroll or medical problems, can hurt trust, cause worry, or even lower how workers feel. The Wall Street Journal (2023) wrote about cases where workers felt tricked instead of trained, which led to anger toward managers.

So, it is morally important to find a balance between being realistic and being sensitive. Simulations should teach, not punish. Being open about what the program wants to do, using anonymous reports, and making the difficulty increase slowly are key to keeping workers' trust. This makes sure that awareness programs make people stronger without causing mental harm or tension in the group.

2.5 Training Approaches: Micro-Learning,Gamification,and Behaviour Shaping

Beyond just using simulations, educating individuals about phishing attacks is starting to incorporate modern instructional methods. These methods align with current learning preferences and are based on our understanding of cognitive processes.

To elaborate, traditional methods of phishing education often relied on generic presentations or infrequent simulated attacks. These approaches frequently failed to produce lasting behavioral changes because they didn't consider the specific ways individuals process and retain information. Modern phishing education is shifting away from this one-size-fits-all model.

One key change is the increased use of interactive learning experiences. Instead of passively receiving information, learners are actively involved in the educational process. This might involve solving phishing-related puzzles, participating in group discussions about real-world attack examples, or competing in gamified scenarios that test their ability to identify and respond to suspicious emails or websites.

Another important element is the concept of personalized learning. Recognizing that individuals have different learning styles and levels of technical knowledge, educators are beginning to individualize the content and delivery of their training programs. This can involve using adaptive learning platforms that adjust the difficulty of the material based on the learner's performance or offering a mix of learning resources such as videos, articles, and interactive exercises to suit different preferences.

Psychological principles, like spaced repetition and active recall, are also being integrated into the design of phishing education programs. Spaced repetition delivers information at increasing intervals over time, which helps to improve long-term retention. Active recall

requires learners to actively retrieve information from memory, which strengthens neural pathways and makes the information more memorable.

A focus on real-world relevance is crucial. Instead of teaching abstract concepts, instruction incorporates of current phishing tactics and case studies. This helps learners to understand the potential consequences of falling victim to an attack and motivates them to take the training seriously. For example, a lesson might analyze a recent phishing campaign that targeted employees in a specific industry, explaining how the attackers crafted their emails and what red flags people should have looked for.

By incorporating these new approaches, we can expect to improve the people's ability to recognize and avoid phishing attacks, ultimately reducing the risk of data breaches and other security incidents.

2.5.1 Micro-Learning for Cybersecurity Knowledge

Small lessons give short, focused information that fits easily into what workers do every day. A study by MDPI in 2022 showed that security ideas taught in 3–5 minute lessons were remembered 27% longer than if they were taught in long, one-hour training sessions. Small lessons are a very good fit for teaching about phishing because they are like real phishing attempts – a quick email and a fast decision. These lessons can quickly tell viewers what to search for as they assess any email. They can also show examples of safe emails so that the viewer know what a safe email looks like.

2.5.2 Using Games as a Strategy to Teach Security

Using game elements like points, leaderboards, and rewards can make security training more interesting. A Study by Alshaikh et al. (2023) shows that using games in training increases how many people want to join and how well they remember what they learned. Some people worry that using games might make the threats seem less serious. To use games well, it's

important to find a good balance – using the games to get people interested while keeping the training serious.By introducing mini games into the training it will enable viewers to want to participate and give a sense of accomplishment when they complete levels.

2.5.3 Changing Behavior through Continuous Reinforcement

What we know about behavior shows that to make lasting change, people need regular reminders and follow-up. This idea includes things like nudges – small reminders that guide people's choices without forcing them. When it comes to phishing, following up might include regular reminders, showing safe examples of what phishing looks like, or asking people to check suspicious emails. A long-term study by Caputo et al. (2022) showed that companies using these reminder saw 60% more phishing reports compared to those that only did yearly training.The follow ups will help keep security top of mind, instead of something that will be forgotten.

2.6 Comparative Analysis of Commercial Simulation Platforms (KnowBe4 and Cofense)

The phishing awareness training sector sees heavy use of commercial platforms. Two major players in this space are KnowBe4 and Cofense. KnowBe4 gives users the ability to change phishing simulations. It also presents data in real time and can work with common compliance rules. Cofense centers its approach on shared information. It uses user-submitted reports from different sectors to keep its threat simulations current.

Both platforms are helpful in big companies, but they have weaknesses:

Cost: The price to subscribe can be too much for smaller groups, such as small schools or charities, that do not have much money.

Complexity: These platforms need users to have a certain level of IT skills, which many smaller places might not possess.

Cultural Relevance: The standard templates might not connect with all local situations. For example, a phishing attempt in a Nigerian school setting might look different. This can make it harder for users to relate to the training.

KnowBe4 and Cofense are great for large businesses. There is still a need for more affordable options that fit different situations. This project, which works with Google Workspace, hopes to meet that need.It aims to develop something that doesn't cost too much and can be adjusted to fit various local needs.

2.7 The Role of Google Workspace in Security Training and Simulation

Google Workspace provides a distinctive and cost-effective platform for phishing simulations because it is very common in educational institutions, non-governmental organizations, and small to medium-sized enterprises. Its components, like Gmail, Google Apps Script, Google Forms, and Google Sheets, can be adapted to develop simple simulation environments.

From a study (BleepingComputer, 2025),It was revealed that attackers are increasingly taking advantage of Google Workspace services, such as creating fake Docs and Drive shares, to carry out credible phishing campaigns. Interestingly, these very same services can also be used for defensive purposes, enabling the creation of secure simulations within the platforms that employees are already familiar with.

Integrating simulations into Google Workspace will make organizations bypass the need to learn external tools and avoid licensing fees. This project aims to take benefit of this by turning a productivity suite into a tool for teaching phishing awareness and defense, making simulation capabilities more accessible to everyone.

The approach centers around modifying current Google Workspace features to realistically mimic phishing attacks. Google Forms can serve as the basis for bogus login pages, while Google Apps Script can automate the distribution of simulated phishing emails and track user

interaction, such as clicks on links or form submissions. The data gathered from these simulations can then be automatically compiled and analyzed using Google Sheets, offering insights into employee vulnerability and areas needing improvement.

One major advantage of this method is its ease of use. Because employees are already comfortable with Google Workspace, they will probably find the simulations less scary and easier to interact with. This can lead to a more effective learning experience, as users are more likely to pay attention and remember the lessons.

Also, integrating phishing simulations into Google Workspace can lead to considerable cost savings. Organizations can avoid the cost of buying and maintaining separate phishing simulation platforms by utilizing their current Google Workspace subscription. This can be especially helpful for small and medium-sized enterprises with limited budgets.

In addition to its practical benefits, using Google Workspace for phishing simulations can also help to foster a culture of security awareness within an organization. By often engaging in simulations, employees grow more aware of the dangers of phishing attacks and better equipped to detect and report suspicious emails or links. This proactive approach to security can considerably lower the risk of successful phishing attacks and safeguard sensitive data.

The project has the potential to significantly improve an organization's cybersecurity posture by using the flexibility and commonness of Google Workspace. By turning everyday productivity tools into instruments for security awareness and training, it makes access to efficient phishing simulation capabilities more democratic and helps to create a more secure digital environment for everyone.

2.8 Ethical and Economic Considerations in Security Training and Simulation

The existing work reveals key areas that need attention:

1. **Cost:** Current platforms often have prices that only big companies can afford, which leaves out smaller groups.

2. **Cultural Relevance:** Phishing simulations often don't take into account different cultural or regional features. This makes them less useful in non-Western areas. To fix this, simulations should be designed carefully to represent the target area to create the most awareness.

3. **Long-Term Maintenance**: There aren't many studies on how awareness programs can be maintained over time, especially in places with limited resources. For continued awareness, there needs to be a focus on how to continue running programs, and keeping people engaged. Budgets should be addressed and models should be put in place to work in low-resource situations.

4. **Use with Everyday Tools:** We don't know much about how to use platforms like Google Workspace for training to raise awareness. Since many use this platform, there should be thought into how it can be used to promote awareness through phishing simulations.

This project plans to deal with these issues by:

1. Creating a cheap simulation platform that works with Google Workspace, designed for NGOs, schools, and small and medium-sized enterprises. The simulation will have an easy-to-use layout. The cost will be looked at so that schools, NGOs, and small and medium-sized enterprises can use it.

2. Adding phishing templates that are culturally relevant and reflect local attack methods. This will make the simulations more realistic. The phishing simulations can be adjusted based on region also.

3. Suggesting an awareness model that is sustainable and balances realism, ethics, and being easy to access. The goal is to address concerns around realism, so the simulation is effective. The simulation must follow ethical guidelines, ensuring that the process doesn't negatively effect the user. The simulation needs to be easy for all groups to get to and use.

CHAPTER THREE

SYSTEM ANALYSIS AND DESIGN METHODOLOGY

3.0 Introduction

This section details the development model for the Phishing Simulation and Awareness Portal. It gives a system analysis and design. The analysis defines the problem, find shortcomings in current methods, and turns stakeholder needs into specific requirements. The design part then connects these requirements to a practical structure using Google Workspace tools (Gmail, Apps Script, Sheets, and Looker/Google Data Studio). All these results in a clear plan that is technically feasible, cost-effective, scalable and can grow in environments with limited resources (NGOs, schools, and SMEs).

3.1 Overview of the Existing Systems

Cybersecurity awareness efforts in many small and medium-sized organizations tend to be informal and react to events. Staff might get occasional alerts by email or at meetings, but this rarely shifts habits for the long term. Most places don't have the tech to run realistic phishing attack simulations or track how users act using data.

If phishing simulation tools are available for purchase, the cost, setup, and integration can be too much for some organizations. Smaller groups, like schools and non-profits, usually stick to simple awareness materials or occasional IT notices, which don't give hands-on learning.

The limits of these old methods include:

Learning materials are not updated: Users see presentations or slides just once, and there are no simulated attacks to reinforce what they learned.

No behavior data: There's no standard way to track vulnerability, click rates, or trend responses.

Keeping records by hand: Tracking participation or incidents often happens in spreadsheets, which causes delays and data problems.

Lost learning chances: If staff click bad links, they do not get automated feedback right away.

Poor process controls: Verification is not consistent for sensitive actions like money transfers or sharing documents.

3.2 Review of the Existing Systems

3.2.1 KnowBe4 Security Awareness and Phishing Simulation Platform

KnowBe4 is a widely adopted, cloud-based platform for security awareness and phishing simulations. It helps organizations improve their defenses by training users and providing feedback based on their behavior. The platform works as a subscription service, combining simulated phishing attacks, user risk evaluations, and educational content.

KnowBe4 is hosted centrally in the cloud, so all simulations, dashboards, and data analysis occur on the company's secure servers. To get started, organizations connect their employee list, typically via Active Directory Sync or SCIM, to automatically add and update user accounts. After setup, administrators can design phishing campaigns using a wide selection of templates. These templates imitate actual phishing attacks, such as fake password reset requests or bogus invoices.

When a campaign starts, the platform sends simulated phishing emails to users. The system monitors how users interact with these emails such as clicking links or reporting them. It then updates each user's risk score as a result. Users who perform poorly in a simulation are automatically enrolled in short training modules, often brief videos or quizzes that explain how to identify threats in the future.

The administrative interface provides analytics that show click rates, report rates, and other key metrics. Managers can use these data points to judge how vulnerable departments are and decide where to focus training efforts. The dashboard also lets organizations compare their performance against others in their industry.

The system's setup and operational needs however, may make it unsuitable for smaller organizations. It needs reliable internet, external data synchronization, and proper user directory management to work well. In addition, because all data is kept in the vendor's cloud, organizations must follow KnowBe4's data policies. These may sometimes conflict with internal organizational requirements.

In terms of pricing, KnowBe4 charges per user. This can be too costly for schools or nonprofits. Also, while it includes good customization options, it still depends greatly on the vendor's infrastructure. This means there are limits to adapting it locally or combining it with existing school systems, like Google Workspace for Education.

Technically, the platform is built on a multi-tier web service design that uses mail delivery APIs, behavior tracking, and web dashboards. It uses secure web protocols (HTTPS/TLS) for communication and browser-based interfaces for users and admins. The phishing simulations use SMTP with DKIM/SPF to avoid being blocked by spam filters. Even with its complexity, the platform can take some time for non-technical users to learn, and system maintenance relies completely on updates from the vendor.

KnowBe4 shows how a phishing simulation setup can drive behavioral change through automation, but its costs and infrastructure needs make it less accessible to smaller institutions that might prefer simpler tools like Google Workspace.

3.2.2 Cofense PhishMe Simulation and Awareness Platform

Cofense PhishMe is an enterprise platform for phishing simulation and security education. It trains users by repeatedly exposing them to simulated phishing attacks and giving instant feedback. Unlike KnowBe4, Cofense works better with corporate security tools such as Security Information and Event Management (SIEM), threat data, and response systems. So, it helps big companies with both education and detection.

The Cofense system includes the PhishMe Simulator, Reporter, Vision, and Triage. The PhishMe Simulator creates and sends out phishing campaigns. Cofense Reporter is a tool for programs like Outlook and Gmail. With it, users send suspicious emails to their security team. The Triage part gathers and checks these emails to spot real phishing threats on the company's network. Vision automates responses and works with threat detection systems to stop bad stuff across the company.

Admins make campaigns with ready-made or custom phishing templates. They set up groups of people to get the emails and plan when to send them. The fake emails go out through the company's email system. The system keeps track of what users do if they open the email, click links, or use Cofense Reporter to report it. The data goes into charts that show things like how easily people get tricked, how fast they report emails, and how well the campaigns work.

The system also uses adaptive learning. Users who get fooled by phishing emails get signed up for lessons or quizzes that match the kind of attack they didn't spot. This helps make the education specific, so users work on their weak points, like scams that create a sense of urgency or ones that steal logins.

Cofense is often hosted on a hybrid cloud. The simulation services are run from the outside, but sensitive user info might be kept on-site, depending on the company's data rules. The

platform uses secure APIs and strong encryption (TLS 1.2+) for all communication and works with email servers like Exchange, Office 365, and Google Workspace APIs.

Even with its good things, this platform may be hard for smaller groups to use. First, setting it up means doing tricky things with email systems, login methods, and installing tools on all users' devices. Second, it counts on the group having a Security Operations Center (SOC), either inside or hired from someone else, that can handle alerts, problems, and user training info. These things cost money and work that smaller groups like schools, charities, and small businesses might not be able to afford. Plus, the costs of subscriptions, setup, and keeping it running can make it not worth it for groups with small budgets.

From the user's view, Cofense gives lots of charts and reporting features, but the layout is made for cybersecurity pros, not regular admins. The many settings and rules might be too much for smaller teams that don't have IT security experts.

Technically, Cofense puts emphasis on matching data and using threat info for feedback, which makes it stand out from simpler tools. It can compare what users do with current threat data to spot connections between user weaknesses and real attacks. But this makes the system complicated and not as flexible for simple uses.

Cofense PhishMe shows how valuable it is to deeply connect human awareness with security automation, from setting up campaigns to checking data in general. But for smaller groups mostly using Google Workspace and having few tech staff, it's still too costly and hard to set up.

3.3 Gap Analysis and Justification for the Proposed System

KnowBe4 and Cofense PhishMe are designed for big companies with strong cybersecurity. They need special tech, licenses, and staff to run well. This makes it hard for smaller groups

like schools and nonprofits to use them, even though these groups are also at risk from phishing.

There's a chance to make something new: a cheap, simple phishing simulation that works with Google Workspace. It would use things people know, like Google Apps Script, Gmail, and Google Sheets, to be easy to use and grow. This system would let groups fake phishing attacks, track what happens, and teach quick lessons right away. It wouldn't need extra subscriptions or be hard to set up.

The Phishing Simulation and Awareness Portal using Google Workspace Tools aims to fix this problem. It makes phishing training available to more people by being automatic, easy to use, and affordable, all in one place.

3.4 Analysis of the Proposed System

This system gives administrators a simple way to run phishing simulations inside Google Workspace. They can make fake phishing campaigns, send simulated emails to employees, and then see who clicks on the bait. If someone falls for the simulation, the system instantly gives them a short training lesson to teach them what they missed. The system also includes dashboards that show how well the simulations are working, tracking progress and pointing out areas where users are getting better at spotting phishing attempts.

To elaborate more on this,it's important to understand why such a system is beneficial. Phishing attacks are a continuous threat to organizations of all sizes. Employees are often the weakest link in a company's security, as they can be tricked into revealing sensitive information or giving access to malicious actors. Regular security awareness training can help defend against these attacks, but traditional methods are not always the most helpful.

This system offers a more practical approach. By running simulated phishing campaigns, administrators can see exactly who is vulnerable and what types of attacks are most likely to

succeed. The short, instant training lessons provide immediate feedback to users, reinforcing the key learning points. This is more impactful than sitting through generic training sessions that may not be relevant to the latest threats.

The dashboards are also a key feature. By tracking performance over time, administrators can see how the training is paying off. They can also identify trends, such as certain departments or individuals who need extra help. This data-driven approach allows them to fine-tune their training programs and make them more successful.

The Google Workspace-native design is another advantage. Since many organizations already use Google Workspace for their email and collaboration needs, this system integrates seamlessly into their existing infrastructure. This makes it easy to set up and use, without requiring extra software or complicated configurations.

This system offers a simple but helpful way to defend against phishing attacks. By running simulated campaigns, giving instant feedback, and tracking performance, it helps organizations improve their security awareness and reduce their risk of a successful attack. This practical approach to security training is more appealing and impactful than traditional methods, ultimately protecting organizations from the growing threat of cybercrime. The ability to measure results and adapt the training based on data also ensures that the security awareness program remains relevant and helpful over time.

3.4.1 Objectives of the Proposed System

* Simulate phishing attempts using Gmail.

* Track and study how users interact with these simulations, noting opens, clicks, and reports.

* Give users instant, relevant feedback to support their learning.

* Supply managers with dashboards for monitoring and data-driven decisions.

. Keep costs down, avoid complexity, and make upkeep simple.

3.4.2 Scope of the Proposed System

This platform is designed to work with email-based simulations and awareness processes inside Google Workspace. It works well for educational institutions, non-profit groups, and small to medium-sized businesses that depend on Gmail and Google Drive.

The system provides a way for these groups to improve their security understanding and practices using tools they already know. For example, schools can use simulated phishing emails to teach students how to spot malicious messages, while businesses could train workers to handle sensitive data responsibly in Google Drive. The idea is to make security training part of their everyday work, rather than an extra burden.

Even though the platform can be expanded, its first release is more interested in training and behavior measurement than in complete system integration or endpoint security. That means it doesn't try to watch every device on the network or connect to big security information systems right away. Instead, it puts learning and behavior changes first.

The platform collects data on how people respond to training exercises, such as how many click on simulated phishing links or how they report suspicious emails and by tracking these metrics, organizations can see how well their training is working and customize it to meet specific needs. Over time, this focus on learning and feedback should lead to a stronger security culture across the entire organization.

Instead of trying to do everything at once, it takes a focused approach. Groups can improve their security knowledge through practical training and clear measurements, making sure everyone knows their part in keeping the organization safe.

3.4.3 Feasibility Study

To determine if a project is feasible, several factors need consideration.

Technical Feasibility : This project depends on Google Workspace specifically, Apps Script, Gmail, and Sheets. A benefit of using these tools is that there's no need to set up or maintain separate servers, because Google handles that. The code is directly written in Google's Apps Script editor, and it can then interact with the Gmail and Google Sheets services without needing external hosting. It supports immediate deployment after coding is finished.

Economic Feasiblity : Many educational institutions and non-governmental organizations already have Google Workspace licenses or can get them at a reduced cost. By using these existing licenses, the project can keep expenses down. The project doesn't have added licensing fees or infrastructure costs.

Operational Feasibility : Since most people are familiar with Gmail, Forms, and Sheets, there should be less resistance to adopting the new system. Staff will already know how to use the basic functions, reducing the training needed and making the transition smoother. It lowers the learning curve and promotes quicker adoption.

Schedule Feasibility : The project is designed in a modular way, so it can be rolled out in stages. A basic working model, which includes email functionality and click logging, can be developed first. Then, more features can be added later, like dashboards and advanced templates. It allows for quicker initial deployment and supports iterative improvements based on user feedback.

3.4.4 Assumptions and Dependencies

- Users have active Google accounts under a Workspace domain.
- Admin has rights to configure Apps Script projects and Gmail sending limits within allowed quotas.
- Organizations will permit monthly simulations and basic reporting aligned with policy and ethics.

3.4.5 Constraints

When developing automated email systems with Gmail and Apps Script, some key points need to be carefully considered.

Gmail has specific sending limits for automated emails, and Apps Script has its own execution time limits. These limits can affect how many emails your system can send and how complex your scripts can be. Understanding these quotas is very important for planning the scale and functionality of your automated processes.

Privacy and ethical standards are crucial. Simulations and data processing must protect sensitive information. It is essential that you avoid gathering personal data unnecessarily and make sure that any reporting is done anonymously or with the least amount of identification possible. This approach protects user privacy and follows data protection regulations and organizational policies.

Network policies might place restrictions on your automated system. You should check if your network blocks link redirects or external content, as this can affect how your emails are delivered and how users interact with them. Being aware of these policies helps you avoid technical problems and make sure your system works smoothly within the existing network environment.

3.4.6 Stakeholder and User Analysis

Administrators and Security Leads: These individuals are responsible for setting up awareness campaigns, checking performance data, and handling security policies. This involves defining the scope of training exercises, assessing the success of these exercises in changing user behavior, and adjusting security protocols based on observed vulnerabilities. They also oversee user access privileges and manage incident responses related to phishing or security breaches.

End Users (Staff, Teachers, and Volunteers): This group participates in the simulated phishing exercises and brief training modules designed to educate them about phishing tactics. They are also encouraged to report any suspicious emails or messages they receive. Their active involvement helps in promptly identifying and addressing potential security threats. The training aims to improve their ability to spot and avoid phishing attempts, thereby reducing the likelihood of successful breaches.

Management and Leadership: This level oversees the overall progress of the security awareness programs, allocates funds or other support for these programs, and approves any changes to existing security policies. They use the data provided by administrators about the program's reach and impact to make informed decisions about resource allocation and policy updates. Their support is critical for maintaining a strong security culture within the organization.

IT Support: This team plays an important role by implementing and maintaining the security settings within Google Admin, such as requiring two-factor authentication (2FA), restricting OAuth apps, and setting secure sharing defaults. They safeguard the technical setup to prevent unauthorized access and data leaks. They also provide solutions to technical

problems that arise during the execution of security measures, ensuring the smooth operation of security protocols across the organization.

3.5 Requirements Specification

3.5.1 Functional Requirements (FR)

These describe what the system must do. They are essential for the phishing simulation tool to work as intended.

FR1: Campaign creation and management. The system shall let administrators make and handle phishing campaigns. They should be able to choose from templates such as password reset, donor request, invoice, or exam pack.

FR2: Simulated email delivery. The system shall send simulated phishing emails through Gmail to specific user groups. The timing of these emails should be randomized to avoid detection.

FR3: Event tracking and logging. The system shall keep track of user actions, such as opening emails, clicking links, and reporting phishing attempts. This data, including timestamps and user IDs, should be saved to Google Sheets, following privacy policies.

FR4: Immediate micro-lessons. If a user clicks on a simulated phishing link, the system should immediately show them a short lesson about the dangers of phishing.

FR5: Admin dashboard. An administrator dashboard should show key metrics. This includes click-through rate (CTR), report rate, the number of users who repeatedly click on phishing links, and the time it takes users to report a phishing attempt.

FR6: Phishing reporting mechanism. The system shall to have a clear way for users to report suspicious emails, such as a Report Phish button or a form.

FR7: Report exporting. The system shall allow administrators to export reports in CSV or PDF format for sharing with management.

FR8: Role-based access. The system should have different access levels for administrators and viewers. Administrators should have full control, while viewers should only be able to see reports.

3.5.2 Non-Functional Requirements (NFR)

Usability: The system has simple workflows, keeping the number of steps low for both admins and users.

Security & Privacy: The system avoids capturing user credentials. Data is kept to a minimum, and access to Sheets and dashboards is controlled.

Reliability: The system shall manage quota limits and retries smoothly.

Performance: A campaign with around 1,000 users finishes inside the organization's sending limits by using batching.

Compliance/Ethics: The policy shall include consent language, and micro-lessons are designed to be non-punitive.

Maintainability: The system can be configured using Sheets, and templates are stored externally for easy updates.

Scalability: The system supports several departments by separating campaigns and sheets.

3.6 Conceptual Design Approach

The project is structured around Object-Oriented Analysis and Design (OOAD) principles, employing an iterative and incremental approach to software delivery. Each iteration focuses on delivering specific, verifiable increments of functionality. The planned iterations include:

1. Implementing the basics for campaign delivery and click tracking. This involves setting up the core systems needed to send out campaigns and record when users click on links within those campaigns.

2. Integrating micro-lessons into the platform. This step adds short, focused educational content.

3. Creating dashboards to view key metrics. The dashboards will provide an interface for users to monitor campaign performance.

4. Adding advanced template options. This iteration expands the range of possible templates for campaigns.

5. Rolling out the system organization-wide. The final planned stage will make the system broadly available.

The use of OOAD is an important part of the project's design. It helps in creating a modular architecture, with components such as templates, dispatchers, loggers, and reporters. This modularity promotes a clear separation of concerns, where each module has a distinct purpose and responsibility. This approach to design makes the system more manageable, easier to maintain, and simpler to extend with new features in the future. Overall, this structure supports code reusability and reduces complexity.

3.7 System Architecture

The system is structured around a four-layer design to ensure it is easy to understand, simple to maintain, and adaptable to change. Each layer has a specific function, contributing to the overall operation of the system.

1. **Presentation Layer**: This is the part of the system that users directly interact with. It includes interfaces like Gmail for displaying emails and banners, specific pages for short

lessons, Google Forms for reporting phishing attempts, and Looker dashboards for data visualization. The main goal here is to create a user interface that is recognizable and requires very little training to use.

2. **Application Layer (Apps Script):** This layer uses Apps Script to manage the different campaigns. It takes care of selecting the appropriate templates, customizing content for individual users, and scheduling when content is sent out. It also handles events through web applications, such as managing click endpoints and processing report forms. In addition, this layer is responsible for triggering short lessons and generating emails.

3. **Data Layer (Google Sheets / Drive):** This layer stores all the necessary data using Google Sheets and Drive. This includes campaign information, lists of recipients, and a library of templates. It also keeps logs of events like when emails are opened, links are clicked, reports are sent, along with their timestamps. Access to this data is controlled through Drive permissions. It is composed of a structured Google Sheet serving as a database with six tabs:

- **config** – system configurations
- **campaigns** – campaign metadata
- **templates** – phishing email templates
- **recipients** – list of targeted users
- **events** – activity log (clicks, reports, completions)
- **micro_lessons** – awareness training materials

4. **Analytics & Feedback Layer; (Looker Studio / Sheets Charts):** This layer focuses on analyzing data and providing feedback using Looker Studio and Google Sheets charts. It

includes key performance indicator (KPI) dashboards that track click-through rates (CTR), report rates, the number of repeat offenders, and the time it takes to report incidents. The layer provides reports that can be exported for management reviews. This allows for a clear view of the system's performance and areas for betterment.

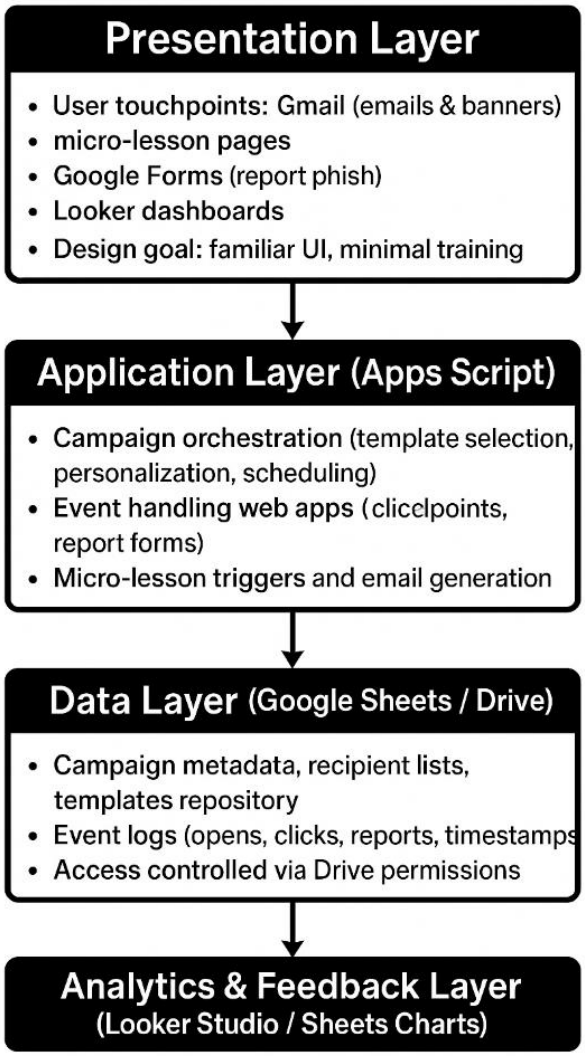

Figure 3.1 System Architecture

3.8 Detailed Module Decomposition

Template Manager: Stores templates, which include subject lines, body text, lure types, and variables.

Recipient Manager: Imports cohorts from spreadsheets and applies segmentation based on department and role.

Campaign Scheduler: Sends emails in batches to stay within Gmail quotas and randomizes sending times.

Mailer Service: Combines templates with user variables and sends emails through Gmail.

Event Logger: A web application that records click events and writes logs to spreadsheets.

Micro-Lesson Engine: Sends educational content immediately after a click and tracks completion.

Reporting & Dashboard: Gathers key performance indicators and allows filtering by date, cohort, and template.

Admin Console: A simple user interface, possibly a Google Sheet menu, for starting jobs and checking status.

3.9 Data Design and Schemas

Sheet: campaigns

- campaign_id, name, template_id, start_date, end_date, status

Attribute	Data Type	Constraint/Description
campaign_id	Integer (INT)	**Primary Key** (PK), Unique identifier
name	Varchar (e.g., Varchar(100))	Campaign display name, **Required**
template_id	Integer (INT)	**Foreign Key** (FK) referencing templates.template_id, **Required**
start_date	Date	The date the campaign begins
end_date	Date	The date the campaign concludes
status	Enum/Varchar (e.g., 'Draft', 'Active', 'Completed')	Current state of the campaign
Attribute	Data Type	Constraint/Description
campaign_id	Integer (INT)	Primary Key (PK), Unique identifier
name	Varchar (e.g., Varchar(100))	Campaign display name, Required
template_id	Integer (INT)	Foreign Key (FK) referencing templates.template_id, Required
start_date	Date	The date the campaign begins
end_date	Date	The date the campaign concludes
status	Enum/Varchar (e.g., 'Draft', 'Active', 'Completed')	Current state of the campaign

Table 3.1 Schema Design (Campaign)

Sheet: recipients

- recipient_id, email, name, dept, role, campaign_id, consent_flag

Attribute	Data Type	Constraint/Description
recipient_id	Integer (INT)	**Primary Key** (PK), Unique identifier
email	Varchar (e.g., Varchar(255))	Recipient's email address, **Unique**
name	Varchar (e.g., Varchar(100))	Recipient's full name
dept	Varchar (e.g., Varchar(50))	Department or organizational unit
role	Varchar (e.g., Varchar(50))	Recipient's job role
campaign_id	Integer (INT)	**Foreign Key** (FK) referencing campaigns.campaign_id, **Required** (links recipient to a specific campaign list)
consent_flag	Boolean (TINYINT/BOOL)	Indicates explicit consent status (e.g., 1=Consented, 0=Opted Out)

Table 3.2 Schema Design(Recipients)

Sheet: templates

- template_id, category (reset/invoice/donor/exam), subject, body_html, micro_lesson_id

Attribute	Data Type	Constraint/Description

Attribute	Data Type	Constraint/Description
template_id	Integer (INT)	**Primary Key** (PK), Unique identifier
category	Enum/Varchar (e.g., 'reset', 'invoice', 'donor', 'exam')	Template type, fixed set of values
subject	Varchar (e.g., Varchar(255))	Email subject line
body_html	Text/LongText	Full HTML content of the email body
micro_lesson_id	Integer (INT)	**Foreign Key** (FK) referencing micro_lessons.micro_lesson_id, Nullable (optional lesson link)

Table 3.3 Schema Design(Templates)

Sheet: events

- event_id, campaign_id, recipient_id, event_type (open/click/report), timestamp, detail

Attribute	Data Type	Constraint/Description
event_id	Integer (INT)	**Primary Key** (PK), Unique identifier (likely auto-incrementing)
campaign_id	Integer (INT)	**Foreign Key** (FK) referencing campaigns.campaign_id, **Required**
recipient_id	Integer (INT)	**Foreign Key** (FK) referencing recipients.recipient_id, **Required**
event_type	Enum/Varchar (e.g., 'open', 'click', 'report')	Type of recipient interaction

Attribute	Data Type	Constraint/Description
timestamp	Datetime/Timestamp	Exact time of the event, **Required**
detail	Varchar (e.g., Varchar(255))	Contextual detail (e.g., the URL that was clicked for a 'click' event)

Table 3.4 Schema Design(Events)

Sheet: micro_lessons

- micro_lesson_id, title, url, duration_sec, tag (urgency/authority/curiosity)

Attribute	Data Type	Constraint/Description
micro_lesson_id	Integer (INT)	**Primary Key** (PK), Unique identifier
title	Varchar (e.g., Varchar(255))	Title of the micro-lesson
url	Varchar (e.g., Varchar(255))	Link to the lesson content, **Required**
duration_sec	Integer (INT)	Length of the lesson in seconds
tag	Enum/Varchar (e.g., 'urgency', 'authority', 'curiosity')	Categorization of the lesson content

Table 3.5 Schema Design(Micro lessons)

3.10 Interface and Interaction Design

Email User Interface: Use short subject lines, real branding, and cues in the email to see if people notice urgency, authority, and scarcity.

Click Landing Page: A safe landing page tracks the click and then shows a short lesson (like, Things you should have noticed).

Report Process: A Report Suspicious link at the bottom sends info to a Google Form or Gmail add-on. These reports are marked as report events.

Admin Console: A Sheet menu (using Apps Script) has buttons to: Make Campaigns, Verify Recipients, Send Emails, See Logs, and Make Dashboards.

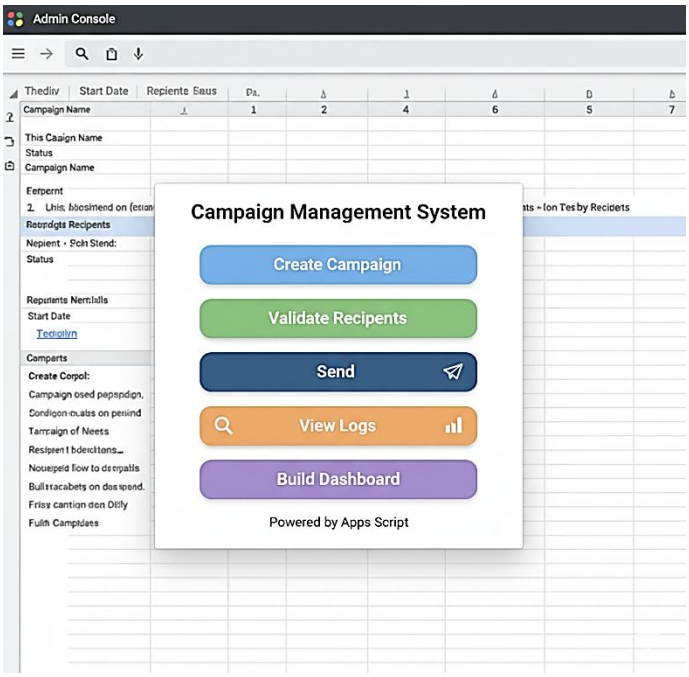

Figure 3.2 Interaction Design

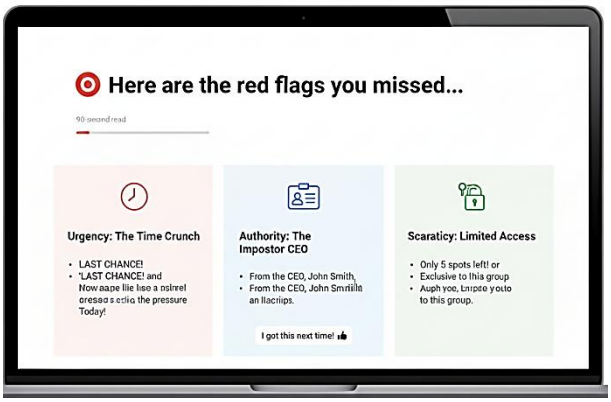

Figure 3.3 Console Mockup Design

3.11 Security, Privacy, and Ethics by Design

Credential collection is prohibited; mock links redirect to secure, internal sites.

Data retention is minimal: logs record only necessary identifiers, and reports can be made anonymous.

Access is restricted: Drive permissions are configured for logs and dashboards, assigning editor/viewer roles as needed.

Participants are informed through policy and training, and a non-punitive strategy is applied.

These practices conform to institutional data policies and relevant privacy regulations.

3.12 UML and Process Models

3.12.1 Use Case Model

Actors: Administrator, End User, and Management.

Use Cases: Campaign setup, sending emails, logging events, delivering micro-lessons, reporting suspicious activity, dashboard viewing, and report exporting.

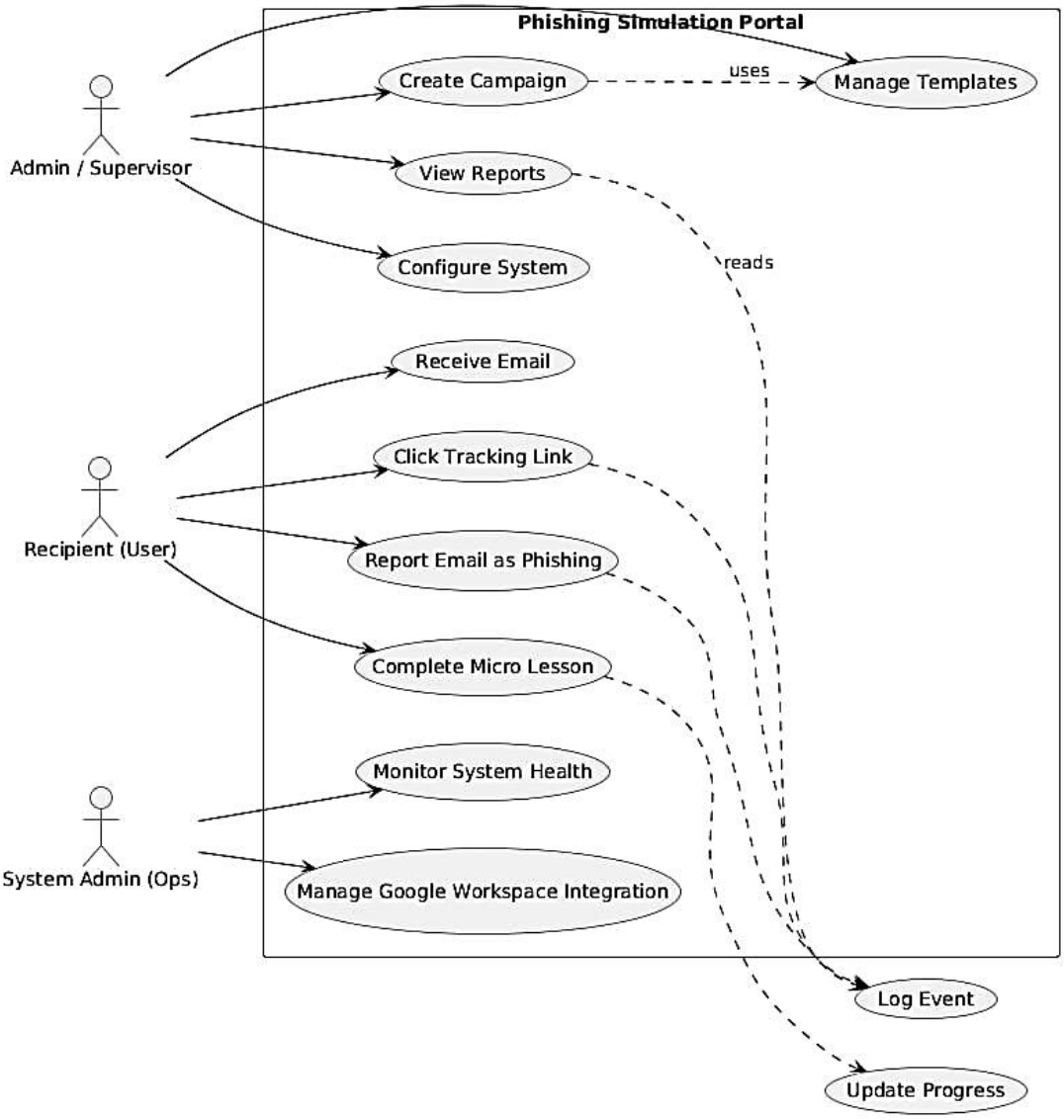

Figure 3.4 Use Case Diagram

3.12.2 Activity Model — Campaign Lifecycle

1. Configure → 2) Schedule → 3) Send → 4) User Interaction (open/click/report) → 5) Micro-Lesson → 6) Aggregate KPIs → 7) Management Review.

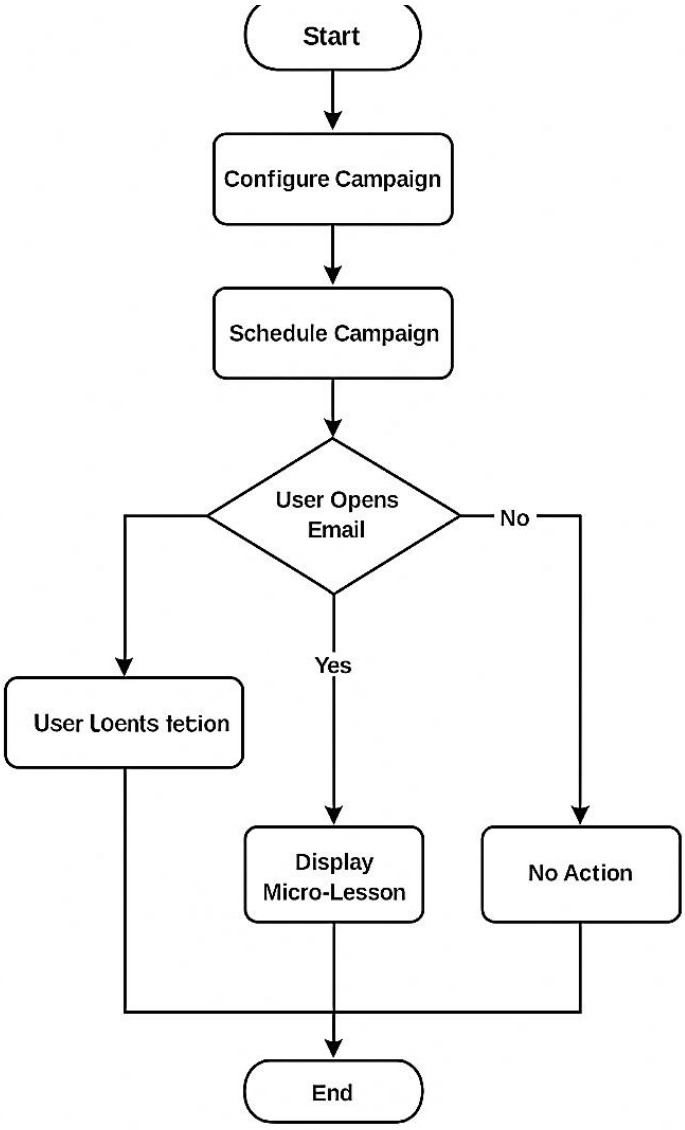

Figure 3.5 Activity Model Diagram

3.12.3 Sequence Model — "Click & Teach"

Admin → Mailer → User → Event Logger → Micro-Lesson Engine → Dashboard

- **Messages**: dispatch(), open(), click(), log(), notifyUser(), aggregate().

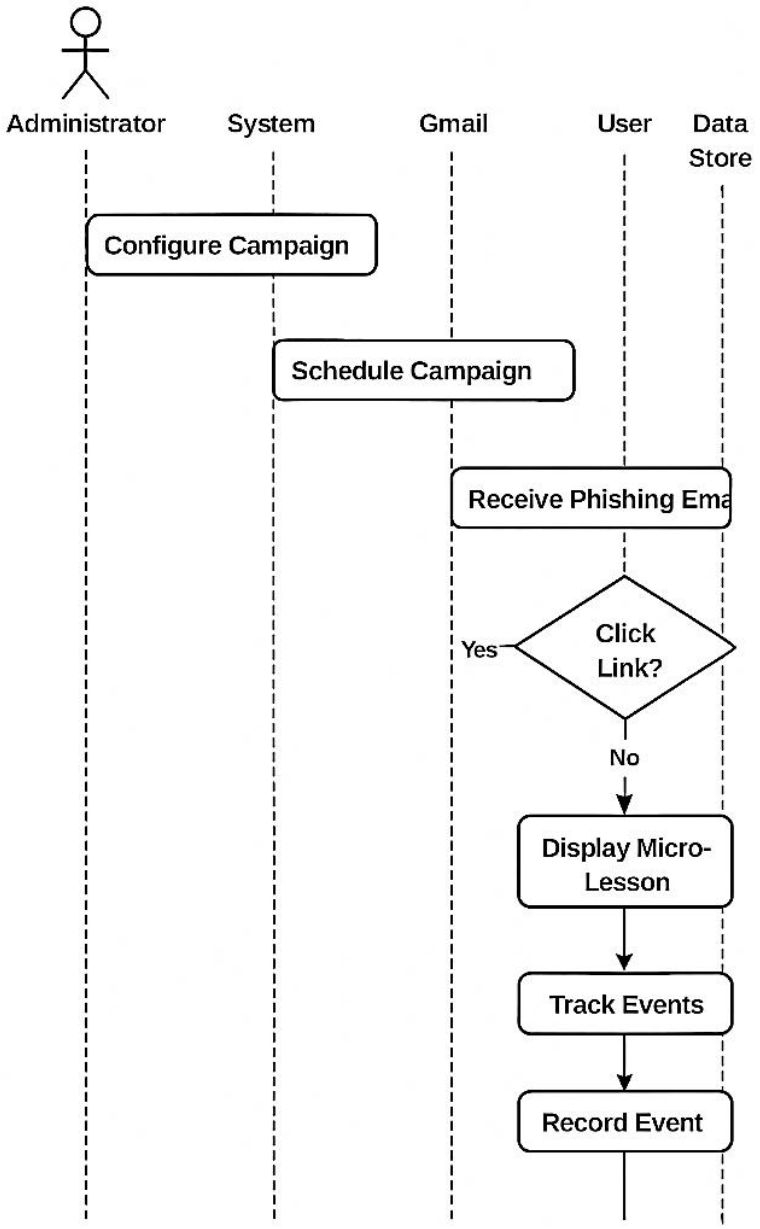

Figure 3.6 Sequence Model Diagram

3.12.4 Class Model (conceptual)

Campaign, Template, Recipient, Event, MicroLesson, DashboardService, MailerService, LoggerService, Scheduler.

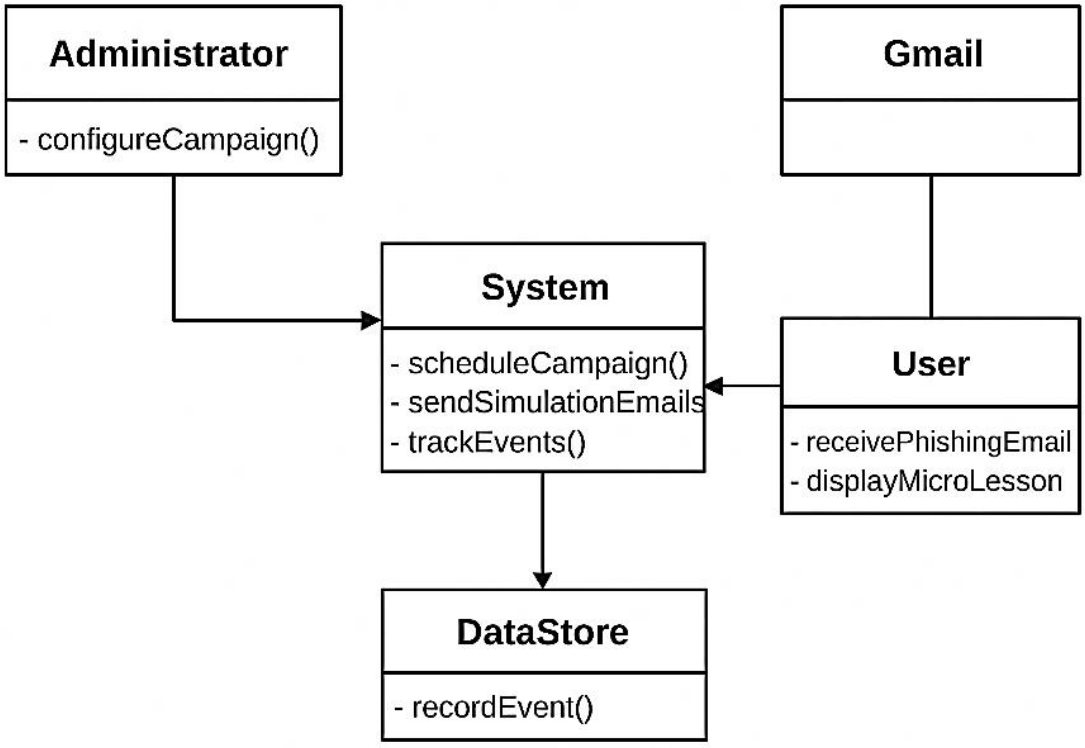

Figure 3.7 Class Model Diagram

3.13 Data Flow Diagrams (DFD)

Context Diagram (Level 0): The organization interacts with Google Services through the portal.

Level 1: Key processes include campaign setup, email dispatch, event logging, micro-lessons, and reporting.

Level 2: Event Logging and Reporting are broken down further into parsing requests, validation, appending logs, and aggregating KPIs.

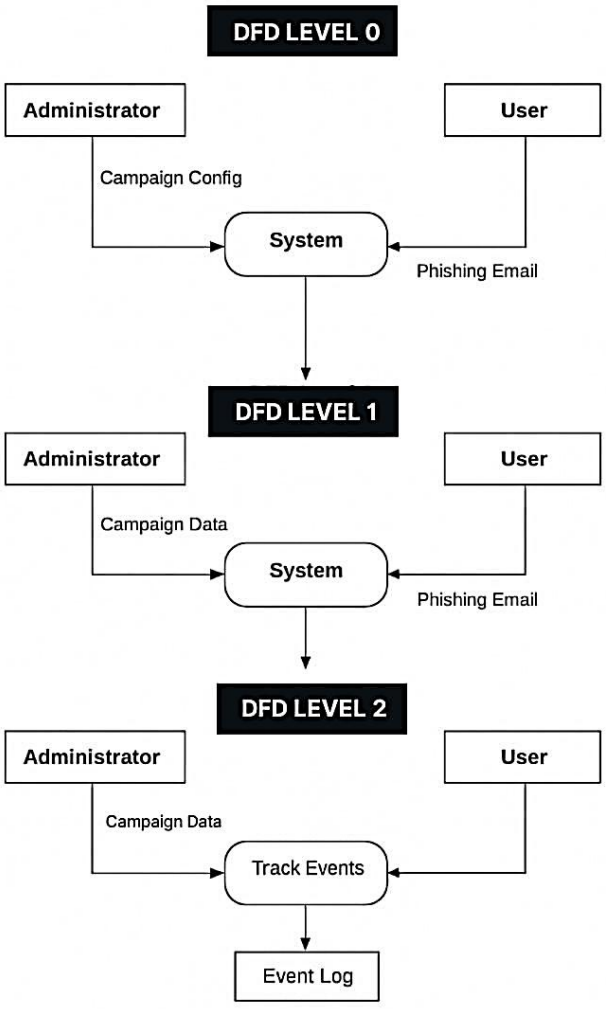

Figure 3.8 Data Flow Levels Diagram

3.14 Core Algorithms and Pseudocode

(A) Batched Dispatch (respect Gmail quotas)

```
for campaign in active_campaigns:
    batch = next_recipient_batch(campaign, batch_size)
    for r in batch:
        msg = render_template(campaign.template, r)
        gmail_send(r.email, msg)
        sleep(jitter())  # randomized delay
```

(B) Event Logging Endpoint (Apps Script Web App)

```
function doGet(e){
  const params = e.parameter;
  appendToSheet('events', [uuid(), params.campaign_id, params.recipient_id, 'click', now(),
params.meta]);
  triggerMicroLesson(params.recipient_id, params.campaign_id);
  return HtmlService.createHtmlOutput(successPageHtml());
}
```

(C) Micro-Lesson Trigger

```
function triggerMicroLesson(recipientId, campaignId){
  const lesson = lookupLesson(campaignId);
```

```
  const email = lookupRecipientEmail(recipientId);
  sendMicroLesson(email, lesson.url, lesson.title);
}
```

3.15 Validation and Verification Strategy

To make sure the system works as expected, several kinds of tests are planned:

Unit Tests:These tests will check whether individual parts of the system, like the template rendering, logger, scheduler, and mailer functions, are working correctly on their own.

Integration Tests:These will test the complete process, from when a notification is sent out to when a user clicks on it, completes a short lesson, and the data is added to the dashboard. This makes sure that all the different parts of the system work together smoothly.

User Acceptance Testing (UAT): Before the system is fully launched, a small group of users (5-10 people) will use it and provide feedback. This feedback will help to improve the clarity and appropriateness of the system.

Measurement : To see how well the system is working, a few things will be measured over a period of 3–6 months. These include:

Click-Through Rate (CTR): the percentage of recipients who click on a link in the reports.

Report Rate: how many reports are submitted.

Time-to-Report: how long it takes for reports to be submitted.

Repeat-Offender Rates: how often the same people are reported more than once.

These measurements will be compared to previous data to see if the new system is making a Positive change.

3.16 Deployment Plan and Change Management

Before a phishing simulation can commence,some things must be done in order. This includes setting up the system, getting the simulation ready, training the users, and watching how well it works as the program goes on.

1. **Environment Readiness**: Start by checking Google Admin settings to make sure two-factor authentication (2FA) is on and that app permissions are set correctly. This ensures the security and integrity of the simulation environment.

2. **Configuration**: Next, fill in the required information in the templates, recipient lists, and campaign sheets. This includes setting up the web-app URL to record the logging endpoint. Careful configuration ensures that the simulations run smoothly and data is captured correctly.

3. **Rollout** : To start, roll out the simulation to only one department. This limited start lets you find and fix any problems before expanding to the entire organization. Before starting the simulation, give users a briefing to explain the goals and process. Then, run a first simulation with easy-to-spot phishing attempts to introduce users to the training without overwhelming them.

4. **Training** : Training should include short sessions covering how to use the Report button and what happens after a simulation. Clear instructions help users understand their role in detecting and reporting phishing attempts.

5. **Monitoring**: After starting the program, check the logs weekly to look for trends and problems. Also, conduct a monthly review of Key Performance Indicators (KPIs) with leadership to assess the program's overall and guide modifications.

3.17 Risk Analysis and Mitigation

Addressing Quota Limits: When systems exceed their allocated usage limits, implement batch processing to group requests and reduce the number of individual calls. In conjunction, apply exponential backoff strategies. This involves progressively increasing the delay between retry attempts after a failure, which can prevent overwhelming the system and improve overall stability. This approach helps to manage demand and avoid service disruptions under heavy load.

Reducing False Alarms and User Concerns: To mitigate false positives, especially when dealing with content that has ethical considerations, it's important to have clear and transparent policies. These policies should clearly outline the criteria for flagging content and the steps taken in response. In place of punitive measures, focus on providing supportive coaching and guidance to users. When a user is flagged due to a possible violation, the emphasis should be on education and improving understanding of the policies. This method reinforces correct practices without discouraging engagement.

Protecting Data from Exposure: To ensure data security, systems should be built to avoid collecting sensitive information. Access to system logs should be strictly controlled, with permissions granted only to authorized personnel who need them for specific tasks. Regular security audits should be conducted to identify and address potential vulnerabilities, making sure that data handling practices comply with established security protocols. These audits include reviews of access controls, data storage methods, and data transmission procedures.

Improving User Participation: To boost participation, change the templates used for various tasks on a regular schedule to keep the user interface fresh and engaging. These templates can be adapted based on the specific role of each user, providing content and options that are relevant to their job. To further motivate reporting and participation, integrate small rewards

or acknowledgments for users who contribute actively. These micro-rewards can be as simple as recognition badges or points that can be redeemed for minor benefits, providing ongoing incentives for user involvement.

CHAPTER FOUR

IMPLEMENTATION AND TESTING

4.0 Introduction

This chapter details the development of a Phishing Simulation and Awareness Portal using Google Workspace. It covers the methods and technologies used to build a working model from the initial system design, the implementation process, how it was tested, and setup of configuration. It centers on building the designed structure, bringing together the user interface, application logic, and data management to meet the goals of simulating phishing attempts and increasing user awareness.

4.1 System Design Tools and Environment

The system was built with Google Workspace tools and Google Apps Script because they work well together, are easy to use, and can grow as needed. They also connect smoothly with Gmail, Sheets, and Forms.

The key development tools and environments are described below:

1. Google Apps Script Environment:

Google Apps Script, a JavaScript platform in the cloud, was used for the main parts of the application. This platform automates tasks and connects different Google services. Scripts were written in this environment to handle email campaigns, save user answers, and start micro-lessons.

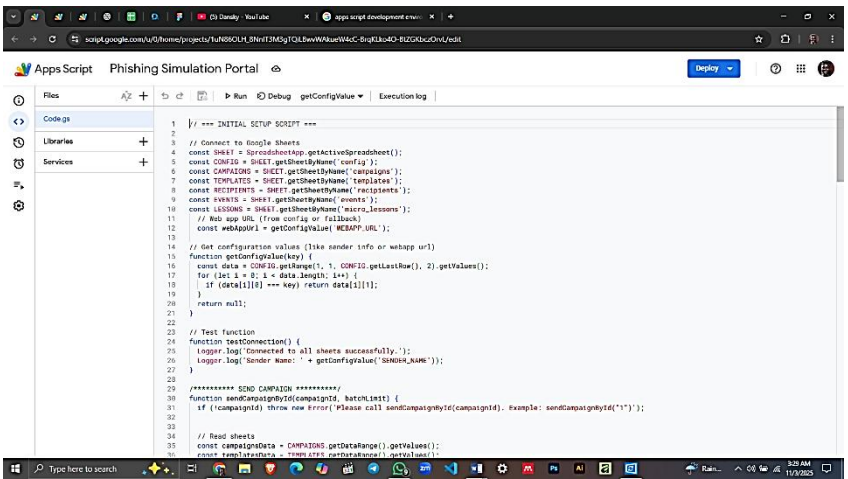

Figure 4.1 A snapshot of the Google Apps Script Environment

2. Google Sheets Database:

The database component was designed using Google Sheets, organized into six structured tabs — **config, campaigns, templates, recipients, events, and micro_lessons**. Each tab performs a distinct role within the system:

config: Contains general configuration parameters such as web app URLs, sender email addresses, and trigger settings.

campaigns: Stores campaign details including campaign ID, subject, and status.

templates: Holds phishing email templates and awareness content.

recipients: Lists all target participants with their email addresses and associated campaign identifiers.

events: Logs user actions such as email opens, link clicks, and reported phishing events.

micro_lessons: Contains the short awareness lessons sent automatically to users after simulated phishing clicks.

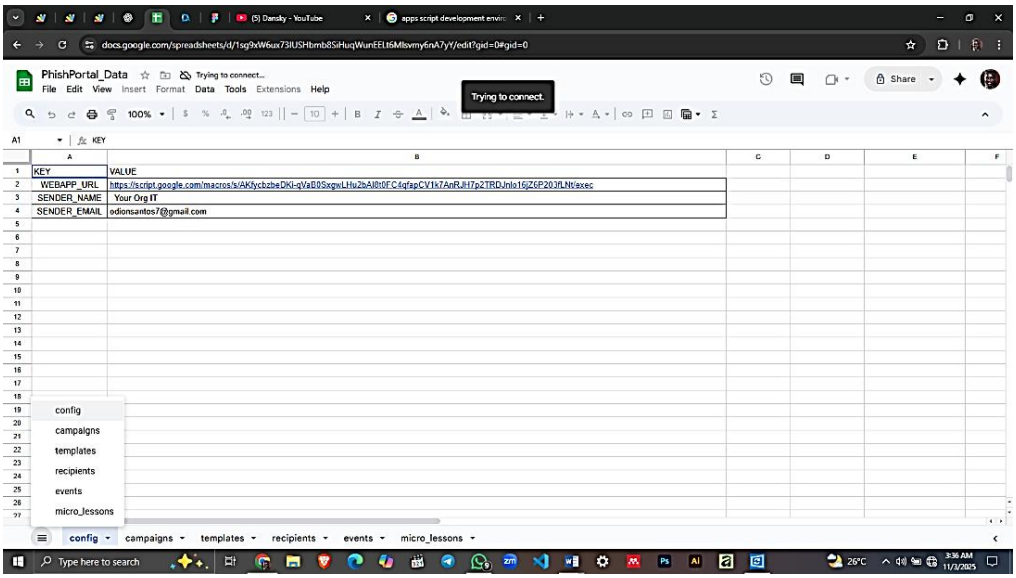

Figure 4.2 A snapshot of the Google Sheets Database Environment

3. Google Forms:

A custom Google Form was configured to capture user reports when they suspect phishing messages. This serves as a simulated "Report Phish" mechanism within the training environment.

4. Google Looker Studio (Data Visualization):

The event data collected through the Google Sheets backend was visualized using Looker Studio dashboards. This interface provides real-time analytics, including user click rates, report statistics, and awareness engagement metrics.

Figure 4.3 A snapshot of Google Looker Studio

5. Hardware and Network Requirements:

The system requires only a standard internet-enabled computer or mobile device with a web browser and access to a Google account. No external server infrastructure was required due to the cloud-hosted nature of Google Workspace.

4.2 System Implementation Process

During setup, the configuration sheet was filled with necessary system details. This included the web application's address, the sender's email, trigger specifications, and API keys. These details act as general references for all system parts. With this setup, the administrator could change important details without needing to change the main script.

To expand, the process of setting up a system often starts with a configuration phase, where key parameters and settings are defined. In this specific case, a configuration sheet was the central point for entering these details. The parameters entered were carefully chosen to give the system the information it needs to run correctly.

The web application URL, for instance, tells the system where the web app is located, letting different parts of the system communicate with it. The sender email address is important for any automated emails the system sends, like notifications or alerts. Trigger settings define when certain actions should occur automatically, making sure the system responds to events

in a timely way. API keys give the system access to external services or data sources, growing its capabilities.

By keeping all these parameters in one place, the configuration sheet acts as a single source of truth for the entire system. This design has some important advantages. First, it makes it easier to manage and update system settings. Instead of having to change code in many different places, an administrator can simply edit the configuration sheet. This reduces the chances of introducing errors and simplifies maintenance.

It also gives flexibility and customization. The administrator can customize the system's behavior by changing the parameters in the configuration sheet. For example, they could change the sender email address, adjust trigger settings, or use different API keys without affecting the system's core code.

This approach of using a configuration sheet with global references is a common practice in software development. It promotes modularity, maintainability, and flexibility, making it easier to build and manage complex systems. The administrator can keep key details separate from the main script and quickly adjust system settings as needed, making sure the system remains adaptable and easy to manage.

4.2.1 Configuration Setup

The configuration phase involved populating the **config** sheet with essential system parameters such as the deployed web app URL, sender email address, trigger settings, and API keys. These parameters serve as global references for all modules of the system. This approach allowed the administrator to modify key details without altering the main script.

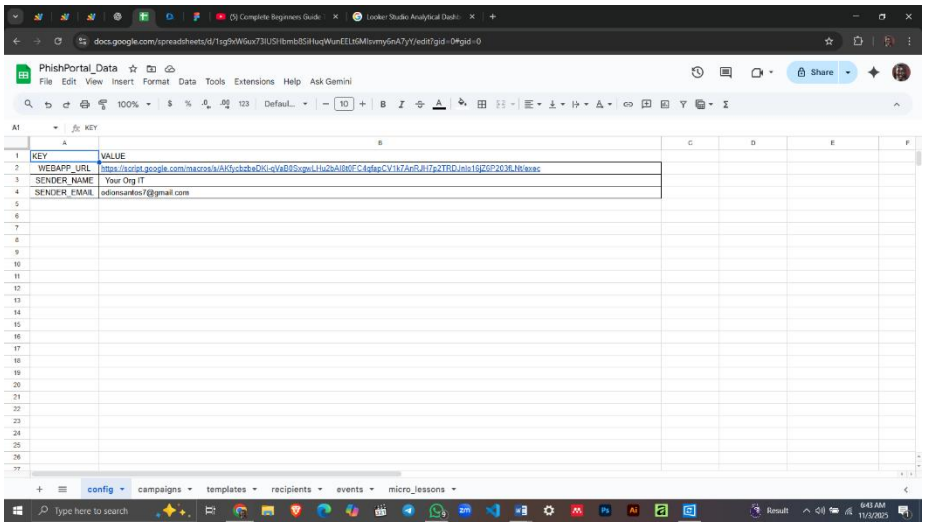

Figure 4.4 Config Module in the Google Sheets Document

4.2.2 Campaign Module

The campaign module allows users to create, change, and schedule phishing simulation campaigns. Each entry in the campaigns tab shows the campaign ID, subject, template, target list, and status (active or completed). After starting, the Apps Script personalizes and sends phishing emails to recipients by campaign template.

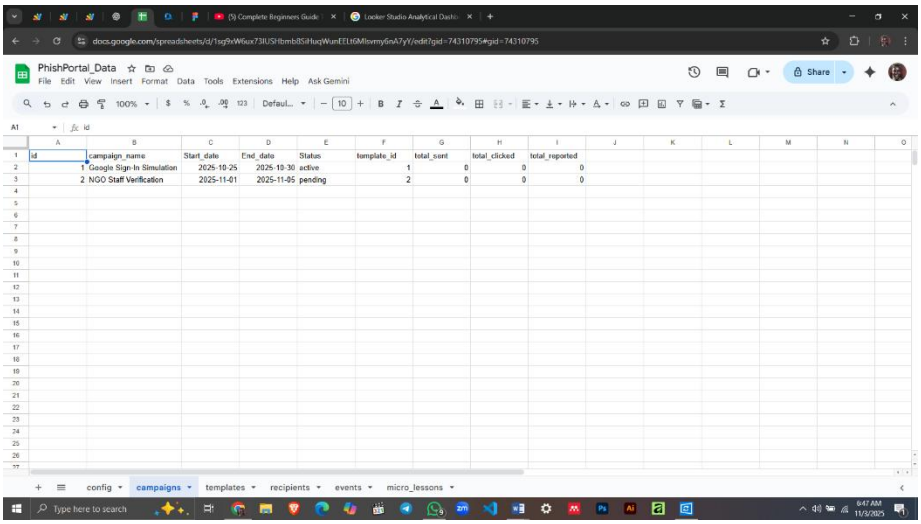

Figure 4.5 Campaign Module in the Google Sheets Document

4.2.3 Event Tracking Module

An event is immediately recorded in the Google Sheet's events tab whenever a receiver responds to a spoof phishing email, such as by opening the email or clicking the embedded link. Details like the campaign ID, timestamp, event type, and recipient email are recorded by the Apps Script web application. Administrators can keep an eye on user behaviour and gauge awareness levels thanks to this real-time surveillance.

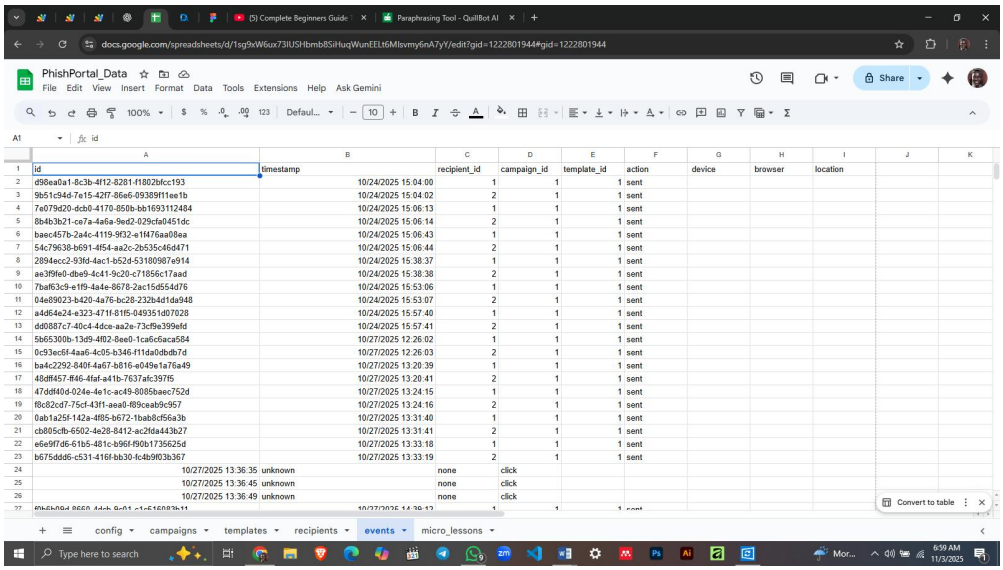

Figure 4.6 Events Module in the Google Sheets Document

4.2.4 Micro-Lesson Module

This module addresses system awareness. When a user clicks a simulated phishing link, they are redirected to a micro – lessons page. This page teaches them how to spot similar phishing attempts. The awareness page offers brief training on email safety and security practices within the organization.

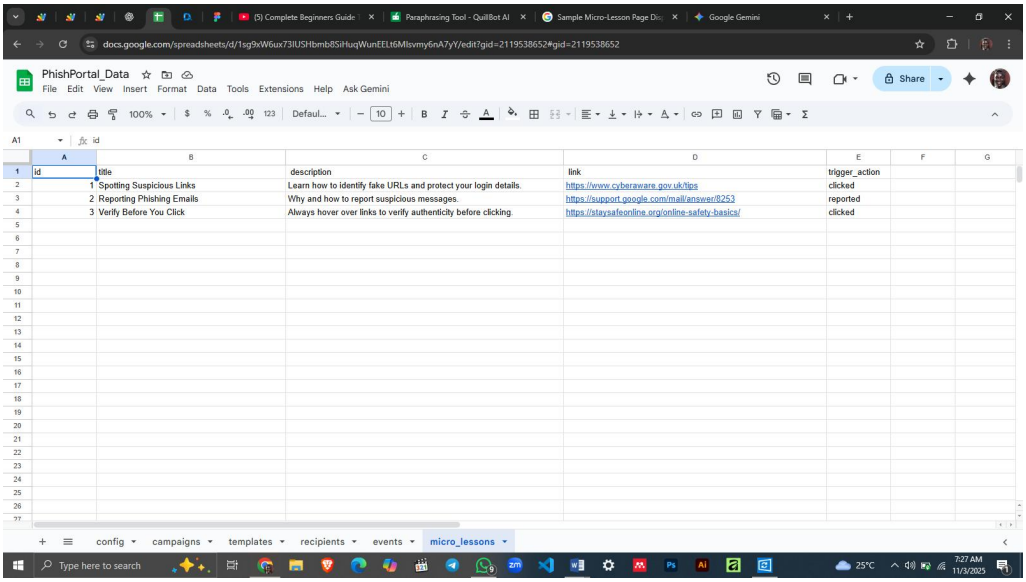

Figure 4.7 MicroLessons Module in the Google Sheets Documents

4.2.5 Reporting and Visualization Module

The system's reporting functionality was implemented using Google Looker Studio, connected directly to the Google Sheet backend. Google Looker Studio was used to make the reporting tool, which links right to the Google Sheet data. The dashboard shows charts and graphs of things like emails sent, clicks, reports done, and how many people finished the training. This gives managers a good idea of how the organization is doing on security.

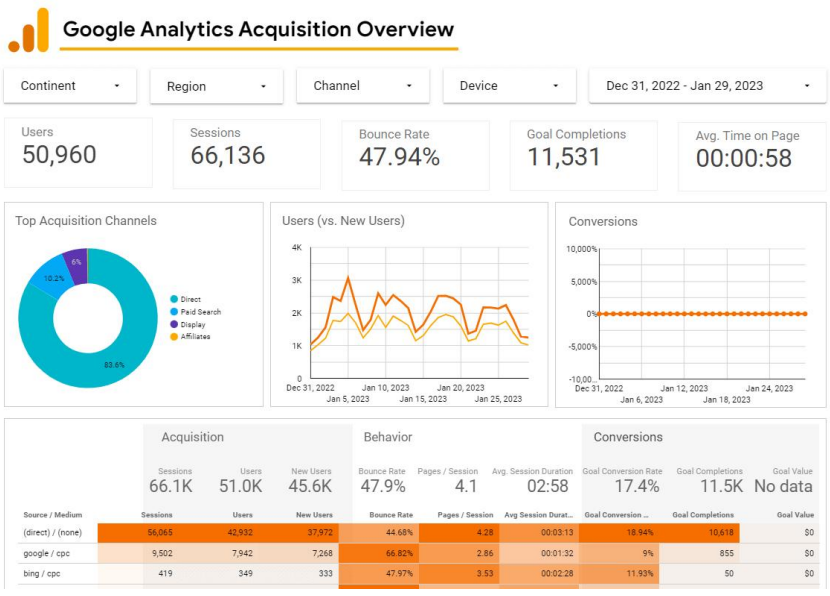

Figure 4.8 A snapshot of the Reporting and Visualisation Module

4.3 System Testing and Validation

Testing was done to check that all parts of the system worked as they should and worked well together. The tests included unit, integration, and user acceptance testing.

1. **Unit Testing**:

Each script function was tested on its own to make sure it did its job right. For example, the sendPhishingEmails() function was tested to confirm it sent emails to the correct people using the right templates.

2. **Integration Testing:**

After unit tests, the way different modules worked together was checked. This meant confirming that clicks from Gmail correctly added info to the events tab, and that related micro-lessons started without needing someone to do it manually.

3. **User Acceptance Testing (UAT):**

A test campaign was started with some participants. Emails were sent without issues, and clicks showed up on the events tab right away. People who clicked the fake phishing links were sent to educational micro-lessons, which proved the system understood its goals.

S/N	Recipient Email	Email Sent	Click Recorded	Report Submitted	Lesson Triggered	Status
1	user1@example.com	Yes	Yes	No	Yes	Passed

S/N	Recipient Email	Email Sent	Click Recorded	Report Submitted	Lesson Triggered	Status
2	user2@example.com	Yes	No	Yes	N/A	Passed
3	user3@example.com	Yes	Yes	No	Yes	Passed
4	user4@example.com	Yes	No	No	N/A	Passed
5	user5@example.com	Yes	Yes	Yes	Yes	Passed

Table 4.1: Sample Test Results of the Phishing Simulation Campaign

4.4 System Deployment

System deployment represents the process of transferring the developed solution from its design and testing environment into a live operational setting where real users can interact with it. In this project, the deployment process was uniquely structured around Google Workspace infrastructure, which provided a cloud-based platform capable of hosting both the backend logic and the frontend user interaction layers without additional hosting costs.

The deployment procedure ensured that the Phishing Simulation and Awareness Portal could be accessed by end users (participants) through Gmail links, while the administrator could manage campaigns and monitor responses through Google Sheets and Looker Studio dashboards.

4.4.1 Deployment Environment

The entire deployment was executed on the Google Cloud-hosted environment. The Google Apps Script platform served as the runtime environment for hosting the web application logic, while Google Sheets acted as the database layer. The choice of this environment offered significant advantages, such as:

• **Seamless Integration**: The system components (Gmail, Sheets, and Forms) communicate natively without the need for third-party APIs or servers.

• **Scalability:** Apps Script automatically scales according to the number of active users without requiring manual server configuration.

• **Security**: Since all activities occur within the Google Workspace domain, user authentication and data access are handled securely through Google's identity management framework.

• **Low Maintenance**: No local installation, manual updates, or hosting fees are required — the system remains fully operational in the cloud.

4.4.2 Deployment Procedure

The deployment was carried out using the Google Apps Script "Deploy as Web App" feature, which generates a public or restricted URL endpoint that users can access when they interact with phishing simulation emails. The following steps summarize the process:

1. Script Publishing:

The completed project scripts were opened in the Apps Script Editor, and the "Deploy as Web App" option was selected. The deployment dialog was configured to ensure that the application executes under the developer's account and that users are allowed to access the web app anonymously (necessary for external recipients).

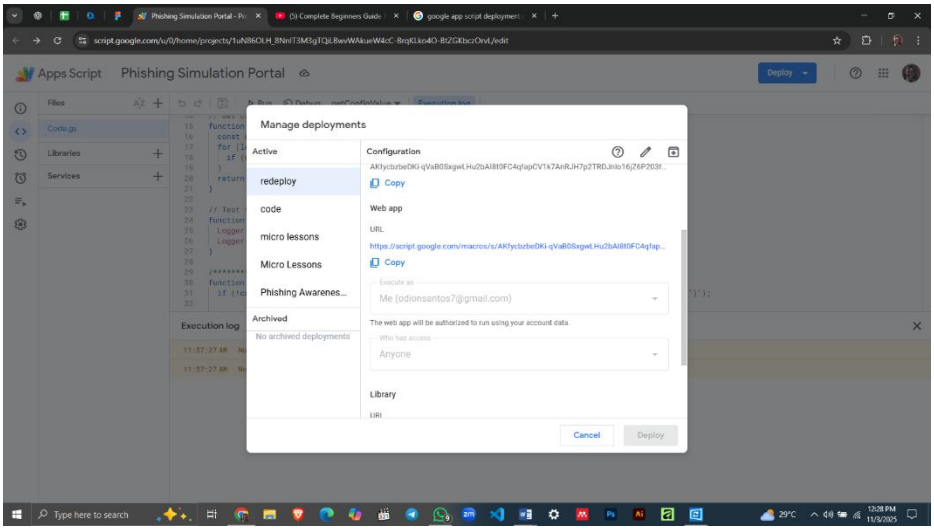

Figure 4.9 Deployment Section of the Google Apps Script Extension

2. Access Permissions Configuration:

To ensure that the script executions run with the necessary permissions, the configuration was set to 'Execute the app as: Me (the developer)' during deployment. The 'Who has access to the app: Anyone' option was chosen so that campaign recipients could trigger the tracking script by clicking on phishing links. This approach protects data integrity and prevents source code modification.

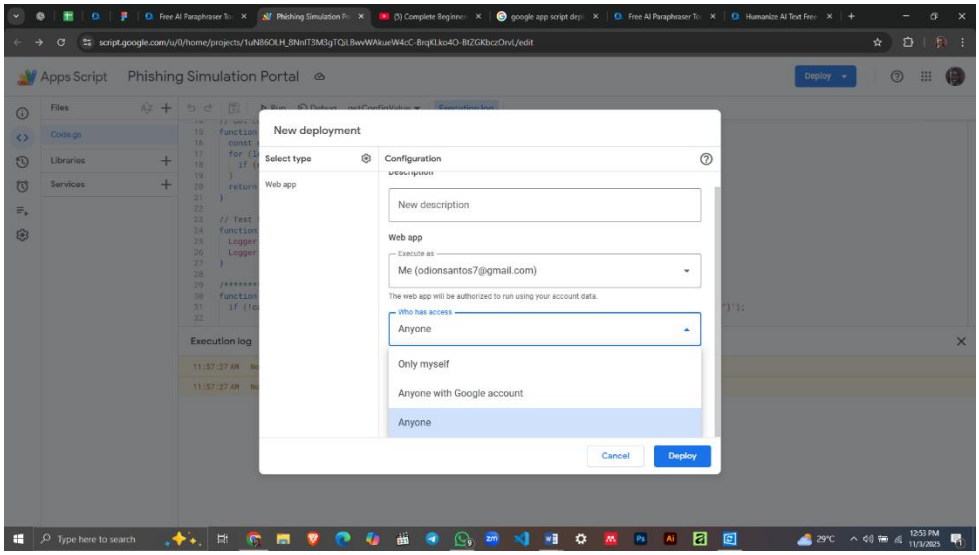

Figure 4.10 Access Permission Segment of the Google Apps Script Extension

2. **URL Generation and Integration:**

Upon successful deployment, Google Apps Script generated a unique web application URL. This URL was then copied and stored in the config sheet as a reference parameter. The same URL was dynamically inserted into the phishing simulation templates through the templates sheet. Consequently, every campaign email sent contained a personalized clickable link pointing to the web app endpoint, which facilitated event tracking and awareness redirection.

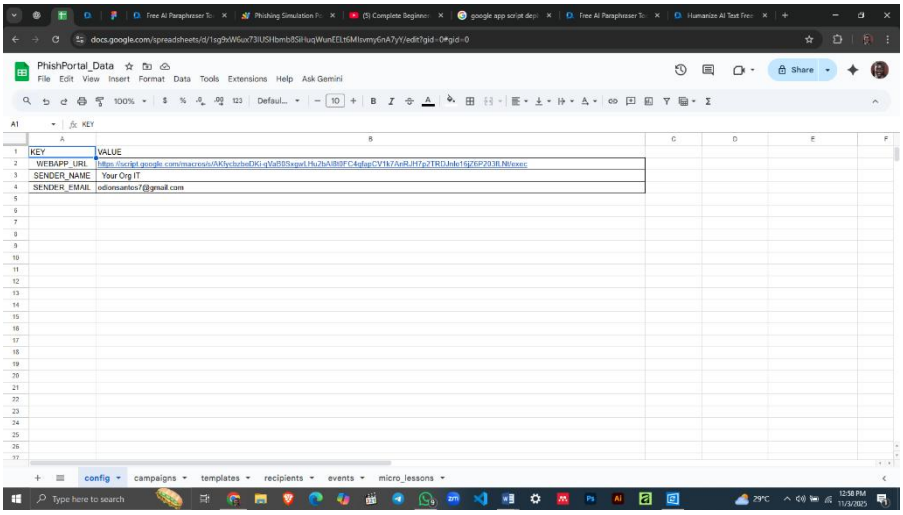

Figure 4.11 Generated Web Url in the Config Module

4. **Database Connection:**

The deployed script communicates with the Google Sheets database using Apps Script's built-in Spreadsheet Service. During testing, all event logs, including link clicks, were successfully transmitted to the events sheet, confirming that the connection between the deployed web app and the database was fully functional.

5. **Trigger Activation :**

The system has automated functions that start processes. These functions include sending scheduled campaigns and creating short awareness lessons. Time-based triggers make sure campaigns run on a regular schedule without an admin having to start them by hand.

To explain more fully, the automated functions operate according to set “triggers.” When these triggers are “activated,” certain processes begin automatically. Consider the scheduled campaigns feature. Instead of an administrator needing to start each campaign at specific times, the system uses a time-driven trigger. The administrator sets the schedule once, and then the trigger makes sure that the campaign launches and sends at the correct times.

Another example is the creation of awareness micro-lessons. These short educational pieces are designed to increase understanding. The system can trigger the auto-generation and distribution of these lessons based on specific events or conditions. The goal here is to provide instant information to users.

The main advantage of these automated triggers is reduced manual workload. The administrator sets up the system initially, defining the triggers and associated actions. After setup, the system handles the repetitive tasks of launching campaigns and sharing lessons. This frees up the administrator to focus on other important work.

Triggers play an important part in making the system self-managing. By automating repetitive processes, the system can run more efficiently and with less human intervention. This automation leads to saved time, reduced errors, and better use of resources. The scheduled campaigns and awareness micro-lessons are just two examples of how trigger activations can improve system operations.

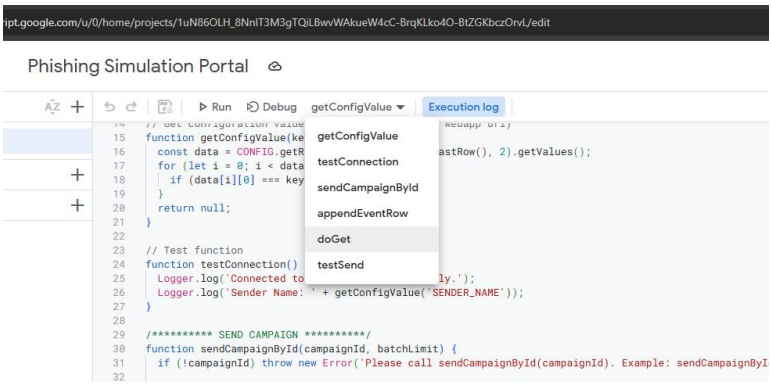

Figure 4.12 Function Section of the Apps Extension

6. Testing the Live Deployment:

Following the deployment of the system, we initiated a pilot test to check its functionality under real-world conditions. This involved sending a test campaign to a selected group of participants. The purpose was to verify that key aspects of the system, like event tracking and content delivery, were working as intended.

As part of the test, when participants opened the emails or interacted with the links included within, the system recorded these actions as events. These events were then immediately logged and accessible for review in the events tab of the system's interface. This immediate logging allowed us to quickly monitor and confirm that the tracking module was accurately capturing user interactions.

For those participants who clicked on the links in the email, they were redirected to a micro-lesson page. This page was hosted on the web application that we had just deployed. The successful redirection and loading of the micro-lesson page served as verification that the awareness delivery module was also functioning correctly. It showed that the system could not only track user engagement but also deliver the intended content without issues.

This pilot test gave hands-on confirmation that both the tracking and awareness delivery modules were operating correctly in the live environment. The real-time event logging and successful content delivery gave confidence in the system's preparedness for broader use.

This testing stage is critical in ensuring a smooth transition from development to full-scale operation, minimizing potential disruptions and ensuring a positive user experience from the outset. The insights gained from this pilot phase will inform any necessary adjustments or refinements before the system is rolled out to a larger audience.

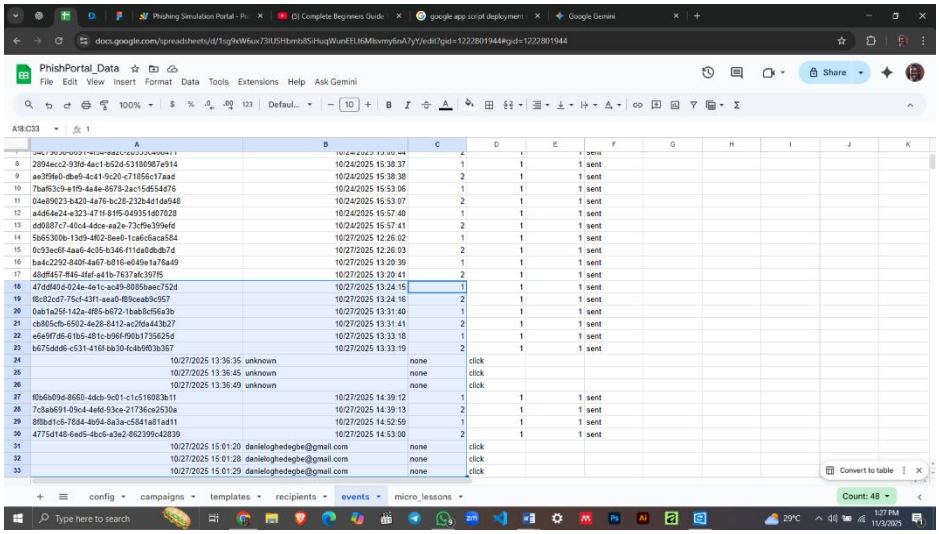

Figure 4.13 Live Events Tracking reflecting on the events Module

4.4.3 Security and Access Control

Security considerations were central to the deployment process. The system leverages Google's built-in authentication and access control mechanisms, meaning that no sensitive data is stored outside Google Workspace.

Only the administrator account has edit permissions on the Sheets database, while participants have view-only or form submission access. Furthermore, Apps Script's execution under the developer's account ensures that only authorized scripts can modify data in the Sheets backend.

The phishing simulation templates were carefully designed to comply with ethical testing standards, ensuring that all recipients were informed afterward that the emails were part of an internal awareness campaign, not a real threat.

4.4.4 Deployment Verification

After deployment, verification was conducted to ensure the system behaved identically in the live environment as it did in the testing environment. This included:

- Verifying that all script functions executed successfully from external email interactions.
- Ensuring that the events sheet accurately recorded timestamps and event types.
- Confirming that the redirection link opened the correct micro-lesson pages.
- Checking that the Looker Studio dashboard updated dynamically with new data.

Each verification test returned positive results, confirming the successful deployment and operational stability of the portal in the production environment.

4.4.5 Maintenance Considerations

The system requires minimal maintenance due to its serverless nature. However, the administrator is advised to periodically:

- Review the config sheet for correct URLs and email sender details.
- Clear old data from the events sheet to maintain performance.
- Re-deploy the script when major updates or new campaigns are introduced.

Because all components are hosted within Google Workspace, version control and updates are handled seamlessly by Google's backend services, ensuring continuous reliability and security.

CHAPTER FIVE

SUMMARY, CONCLUSION AND RECOMMENDATIONS

5.0 Introduction

This chapter provides a comprehensive summary of the entire project work, drawing together the objectives, methods, implementation processes, and key achievements of the Phishing Simulation and Awareness Portal. It also presents the conclusions reached from the research and practical implementation, as well as recommendations for future improvements and possible extensions of the system. The purpose of this chapter is to reflect on the accomplishments of the project in relation to its initial goals and to propose ways it can be sustained and scaled for broader organizational use.

5.1 Summary

The project was conceived to address the growing challenge of phishing attacks that exploit human vulnerabilities rather than technological loopholes. The study recognized that most cybersecurity breaches occur due to lack of user awareness, hence the need to develop a simulation-based training system that helps users identify and respond correctly to phishing threats.

Chapter One discussed the background of the study, highlighting the importance of user education in mitigating phishing-related risks. The problem statement emphasized that while several organizations employ technical controls such as firewalls and spam filters, employees remain the weakest link in the security chain. Therefore, this project aimed to complement existing technical defenses with behavioral awareness through simulated training.

Chapter Two reviewed related literature and existing systems, showing that while various phishing awareness tools exist, many are either expensive, complex to manage, or require

external hosting. The review established the research gap that motivated this study ; the need for a lightweight, low-cost, and integrated phishing simulation system using Google Workspace tools.

Chapter Three detailed the system analysis and design. It described the system's architecture structured into three main layers: presentation, application, and data layers. The presentation layer included user interfaces such as Gmail and Google Forms; the application layer comprised Google Apps Script handling automation logic; and the data layer was implemented with Google Sheets serving as a database. The design also included diagrams such as the Data Flow Diagram, System Architecture, and Flowcharts, which illustrated how information flows within the system.

Chapter Four presented the practical implementation of the system. It explained how Google Apps Script, Google Sheets, and Looker Studio were used to create an interactive simulation and awareness environment. The chapter discussed each implementation step, from configuration setup to deployment, including details of event tracking, campaign management, and automated awareness generation. The deployment on Google's cloud platform ensured accessibility, scalability, and zero maintenance overhead. System testing and validation confirmed that the developed portal performed according to its specifications.

The project in summary,successfully achieved its objectives by developing a cloud-based phishing simulation and awareness portal that integrates email campaigns, event tracking, and micro-lesson delivery into a unified platform. The system effectively demonstrates that phishing awareness can be achieved through practical, simulated experience rather than theoretical training alone.

5.2 Conclusion

The outcome of this research has demonstrated that technology-enabled awareness training can significantly improve user vigilance against phishing attacks. By leveraging Google Workspace tools, the project has successfully implemented a low-cost, accessible, and automated phishing awareness platform that can be deployed in educational, corporate, and non-profit organizations without the need for specialized technical infrastructure.

The Phishing Simulation and Awareness Portal not only meets the project's objectives but also introduces a scalable model that can be expanded for organizational use. It integrates simulation, data analytics, and user feedback into one functional system,thereby promoting a culture of cybersecurity mindfulness.

In essence, this project bridges the gap between theoretical cybersecurity education and real-world experiential learning by using simulation as a teaching mechanism. It also provides evidence that robust awareness systems can be built using existing cloud-based tools, without the need for proprietary software or expensive infrastructure.

The successful deployment and testing of the system affirm its viability as a sustainable awareness solution. It demonstrates the importance of empowering users with continuous training and feedback mechanisms, ultimately contributing to a reduction in phishing susceptibility rates within organizations.

5.3 Recommendations

Based on the development and implementation experience, the following recommendations are made for future improvements and broader application of the system:

1. **Integration with Learning Management Systems (LMS):**

Future versions of the system could be extended to integrate with platforms like Moodle or Google Classroom. This would allow administrators to track users' progress and assign structured awareness lessons as part of formal cybersecurity training programs.

2. **Enhanced Analytics and Reporting:**

   While the current version uses Google Looker Studio for visualization, incorporating machine learning analytics could provide predictive insights such as identifying users most at risk of phishing or measuring behavioral improvements over time.

3. **Multi-Language Support:**

   Adding localization features would make the system adaptable for organizations with diverse linguistic backgrounds. This would increase accessibility and effectiveness in multilingual environments.

4. **Mobile Optimization:**

   Since many phishing attacks occur via mobile devices, the portal could be further optimized for mobile browsers to improve user experience and lesson accessibility.

5. **Integration with Organization-wide Email Gateways:**

   For corporate environments, the system could be enhanced to interface directly with enterprise email gateways (such as Microsoft Exchange or custom APIs) to conduct broader campaigns within official domains.

6. **Continuous Awareness Campaigns:**

It is recommended that organizations using this system run regular phishing simulations (monthly or quarterly) to reinforce user vigilance and measure progressive improvement in cybersecurity awareness levels.

7. **User Feedback and Gamification:**

   Incorporating short quizzes, feedback options, or gamified scoring systems after each micro-lesson could increase engagement and retention of awareness lessons.

5.4 Contribution to Knowledge

This project contributes to the body of knowledge in cybersecurity awareness by demonstrating a practical and scalable framework for phishing simulation using freely available Google Workspace tools. The project shows how simple automation platforms can be helpful for teaching and training about cybersecurity. It gives schools and other organizations a pattern they can follow without spending a lot of money or needing very specialized tech skills, unlike the pricey and complicated solutions that are usually available.

To expand, in element , cybersecurity awareness is one of the most important defenses an organization can maintain to prevent data breaches and intrusions. Human error continues to be a primary cause of successful phishing attacks, making awareness training crucial. Typical training programs, even those that are computer-based, may be expensive, hard to maintain, and sometimes hard to adapt to the changing risk environment. This project tackles these issues by offering a low-cost, simple-to-use option that takes advantage of Google Workspace's widespread availability in learning and professional situations.

This project describes how to set up a phishing simulation system using tools like Google Apps Script, Google Sheets, and Gmail. These tools let users design and automate fake phishing emails, track who clicks on links, and give custom feedback or training based on people's actions. Because everything is done inside the Google Workspace environment, it

removes the need for extra software or hardware, greatly lowering setup and running expenses.

One of the main strengths of this framework is how easily it can change. The simulation emails, training information, and reports can all be changed to match different risks and learning objectives. This makes sure that the training stays relevant and engaging for users. Also, the project emphasizes how important it is to protect people's privacy and act ethically when running these simulations, such as getting permission from participants and making the goals of the training clear.

This project lets institutions of all sizes improve their cybersecurity training without spending a lot of money by providing a realistic example and clear instructions,. It also helps teachers and trainers use low-code platforms to create engaging learning experiences, encouraging a more knowledgeable and secure digital environment. The project's results will be helpful for schools, small businesses, and any group wanting to improve their cybersecurity posture in a budget-friendly way.

5.5 Limitation of the Study

The major limitation of this project is its dependency on Google Workspace infrastructure, which may restrict access in organizations not using Google accounts. Moreover, the system’s automation relies on Apps Script execution quotas, which could limit scalability in very large organizations. Despite these constraints, the system remains a viable proof of concept for small and medium-sized institutions.

.

REFERENCES

Alkhalil, Z., Hewage, C., Nawaf, L., & Khan, I. (2021). Phishing attacks: A recent comprehensive study and a new anatomy model. *Electronics*, 10(13), 1607. https://doi.org/10.3390/electronics10131607

Anderson, R., Barton, C., Böhme, R., Clayton, R., van Eeten, M. J., Levi, M., & Savage, S. (2022). Measuring the cost of cybercrime revisited. *Journal of Cybersecurity*, 8(1), taab026. https://doi.org/10.1093/cybsec/taab026

BleepingComputer. (2025). *Hackers exploit Google Apps Script for phishing campaigns impersonating Workspace*. Bleeping Computer. https://www.bleepingcomputer.com/news/security

Catal, C., Tekinerdogan, B., & Akbulut, A. (2025). Artificial intelligence–enhanced phishing attacks and detection systems: A systematic literature review. *Computers & Security,* 144, 103058. https://doi.org/10.1016/j.cose.2024.103058

Deloitte. (2024). *Global Cyber Trust Insights Survey 2024: Managing the human risk factor.* Deloitte Insights. https://www.deloitte.com/insights

Jansson, K., & von Solms, R. (2022). Phishing awareness training: A behavior shaping approach using feedback and repetition. *Information & Computer Security, 30*(2), 229–243. https://doi.org/10.1108/ICS-11-2021-0124

Kumar, R., & Chatterjee, M. (2023). Simulation-based phishing awareness and training in organizations: Evaluating behavioral change. *International Journal of Information Management*, 69, 102662. https://doi.org/10.1016/j.ijinfomgt.2023.102662

Lain, D., Williams, P., & Fafinski, S. (2024). Reassessing phishing resilience: The effects of targeted simulations and personalized feedback. *Computers & Security*, 132, 103403. https://doi.org/10.1016/j.cose.2023.103403

Kim, S., Lee, J., & Choi, M. (2022). Effectiveness of phishing simulation training on cybersecurity awareness and behavioral change. *Computers & Security*, 118, 102744. https://doi.org/10.1016/j.cose.2022.102744

Naqvi, S. A. R., Memon, R. A., & Qureshi, M. A. (2023). A taxonomy of phishing techniques and countermeasures in cloud-based ecosystems. *IEEE Access, 11*, 43245–43261. https://doi.org/10.1109/ACCESS.2023.3261028

Nguyen, Q. H., & Hall, J. (2024). Gamified security awareness: How microlearning influences phishing detection performance. *Journal of Information Security and Applications*, 80, 103602. https://doi.org/10.1016/j.jisa.2024.103602

Sheng, S., Holbrook, M., Kumaraguru, P., Cranor, L. F., & Downs, J. (2022). Who falls for phishing? A demographic analysis of phishing susceptibility and effectiveness of interventions. *Human Factors*, 64(1), 78–95. https://doi.org/10.1177/00187208211002051

Symantec. (2023). *Internet security threat report*. Symantec Corporation.

Verizon. (2023). *Data Breach Investigations Report*. Verizon Enterprise.

Wall Street Journal. (2023). *When phishing training goes too far: Companies rethink aggressive simulations*. Wall Street Journal. https://www.wsj.com/articles/phishing-training-backlash-2023

World Economic Forum. (2024). *Global Cybersecurity Outlook 2024: Building digital trust in a fragmented world*. World Economic Forum. https://www.weforum.org/reports/global-cybersecurity-outlook-2024

APPENDIX

FEATURE SOURCE CODES

Main project — Code.gs

```
// ========== Code.gs (Main PhishPortal Project) ==========

// Bound to the Google Sheet (PhishPortal_Data)

// Required sheet tabs (exact names): config, campaigns, templates, recipients, events, micro_lessons

// ---------- Sheet connections ----------

const SS = SpreadsheetApp.getActiveSpreadsheet();

const CONFIG = SS.getSheetByName('config');

const CAMPAIGNS = SS.getSheetByName('campaigns');

const TEMPLATES = SS.getSheetByName('templates');

const RECIPIENTS = SS.getSheetByName('recipients');

const EVENTS = SS.getSheetByName('events');

const LESSONS = SS.getSheetByName('micro_lessons');

// ---------- Helper: read config key (single source of truth) ----------

function getConfigValue(key) {

  if (!CONFIG) return null;

  const vals = CONFIG.getRange(1,1,CONFIG.getLastRow(),2).getValues();

  for (let i=0;i<vals.length;i++){

    if (String(vals[i][0]).trim() === String(key).trim()) return vals[i][1];

  }

  return null;

}
```

```
// ---------- Utility: render template with {{placeholder}} tokens ----------
function renderTemplate(templateBody, data) {
  let out = String(templateBody || '');
  for (const k in data) {
    const token = new RegExp('{{' + k + '}}', 'g');
    out = out.replace(token, data[k]);
  }
  return out;
}
// ---------- Append event row to events sheet ----------
// Expected events sheet headers (first row):
// id | timestamp | recipient_id_or_email | campaign_id | template_id | action | device | browser | location | extra
function appendEventRow(eventObj) {
  // eventObj: {recipient: '', campaignId:'', templateId:'', action:'', extra:''}
  if (!EVENTS) return;
  const row = [
    Utilities.getUuid(),
    new Date(),
    eventObj.recipient || '',
    eventObj.campaignId || '',
    eventObj.templateId || '',
    eventObj.action || '',
    eventObj.device || '',
    eventObj.browser || '',
    eventObj.location || '',
```

```
    eventObj.extra || ''
  ];
  EVENTS.appendRow(row);
}
// ---------- Send campaign by id ----------
// campaignId: string or number that matches campaigns.id column
// batchLimit: optional number to limit sends during tests
function sendCampaignById(campaignId, batchLimit) {
  if (!campaignId) throw new Error('Usage: sendCampaignById(campaignId, optionalBatchLimit)');
  // use WEBAPP_URL for tracking (if set), and MICROLESSONS_URL optionally for direct lesson link
  const webAppUrl = getConfigValue('WEBAPP_URL') || '';
  const microLessonsUrl = getConfigValue('MICROLESSONS_URL') || '';
  // read sheets
  if (!CAMPAIGNS || !TEMPLATES || !RECIPIENTS) throw new Error('One or more required sheets are missing.');
  const campaignsData = CAMPAIGNS.getDataRange().getValues();
  const templatesData = TEMPLATES.getDataRange().getValues();
  const recipientsData = RECIPIENTS.getDataRange().getValues();
  const campHeader = campaignsData[0].map(h => String(h).trim());
  const tmplHeader = templatesData[0].map(h => String(h).trim());
  const recHeader = recipientsData[0].map(h => String(h).trim());
  // find campaign row
  const campaignRow = campaignsData.slice(1).find(r => String(r[campHeader.indexOf('id')]) === String(campaignId));
  if (!campaignRow) throw new Error('Campaign id not found in campaigns sheet: ' + campaignId);
```

```
  const templateId = campaignRow[campHeader.indexOf('template_id')];

  const templateRow = templatesData.slice(1).find(r => String(r[tmplHeader.indexOf('id')])
=== String(templateId));

  if (!templateRow) throw new Error('Template id not found for campaign: ' + templateId);

  const subjectTemplate = String(templateRow[tmplHeader.indexOf('subject')] || 'Security
Notice');

  const bodyTemplate = String(templateRow[tmplHeader.indexOf('body')] || 'Hello {{name}},
please verify: {{redirect_url}}');

  // find recipients with status = active

  const recipients = recipientsData.slice(1).filter(r => {

    const st = String(r[recHeader.indexOf('status')] || '').toLowerCase();

    return st === 'active' && String(r[recHeader.indexOf('email')] || '') !== '';

  });

  if (recipients.length === 0) {

    Logger.log('No active recipients found.');

    return;

  }

  const limit = (typeof batchLimit === 'number' && batchLimit > 0) ? Math.min(batchLimit,
recipients.length) : recipients.length;

  Logger.log('Preparing to send to ' + limit + ' recipients (out of ' + recipients.length + ').');

  for (let i=0;i<limit;i++){

    const r = recipients[i];

    const recipientId = String(r[recHeader.indexOf('id')] || ('r' + (i+1)));

    const toEmail = String(r[recHeader.indexOf('email')] || '').trim();

    const name = String(r[recHeader.indexOf('name')] || '');

    // tracking redirect: build URL to main web app that logs click and shows awareness page

    // the main webapp doGet expects: ?email=...&campaignId=...
```

```
    const trackingUrl = webAppUrl ?
${webAppUrl}?email=${encodeURIComponent(toEmail)}&campaignId=${encodeURICom
ponent(campaignId)} : '';

    // personalized HTML body (replace placeholders)

    const personalData = {

      name: name,

      redirect_url: trackingUrl,

      tracking_url: trackingUrl,

      email: toEmail,

      recipient_id: recipientId,

      campaign_id: campaignId

    };

    const htmlBody = renderTemplate(bodyTemplate, personalData);

    try {

      GmailApp.sendEmail(toEmail, subjectTemplate, '', { htmlBody: htmlBody });

      Logger.log('Sent to ' + toEmail);

      appendEventRow({ recipient: toEmail, campaignId: campaignId, templateId: templateId,
action: 'sent', extra: '' });

    } catch (err) {

      Logger.log('Failed to send to ' + toEmail + ': ' + String(err));

      appendEventRow({ recipient: toEmail, campaignId: campaignId, templateId: templateId,
action: 'send_failed', extra: String(err) });

    }

    Utilities.sleep(900); // small throttle to avoid quota bursts

  }

  Logger.log('Send loop finished.');

}
```

```
// ---------- test helper ----------
function testSend() {
  // example: sendCampaignById('1', 2) to send to first 2 active recipients
  sendCampaignById('1', 2);
}
// ---------- Web App: doGet(e) - Awareness Page + Click Logger ----------
// This function is invoked when a user clicks the tracking link in the email.
// Expected query params: email and campaignId
function doGet(e) {
  const params = e.parameter || {};
  const email = params.email || 'unknown';
  const campaignId = params.campaignId || 'none';
  const timestamp = new Date();
  // Log click event to events sheet
  appendEventRow({ recipient: email, campaignId: campaignId, templateId: '', action:
'clicked', extra: '' });
  // Get micro lessons URL from config if available (button will redirect to this)
  const microLessonsUrl = getConfigValue('MICROLESSONS_URL') || '';
  // Build awareness HTML (clean, corporate style) and inject microLessonsUrl safely
  const html = HtmlService.createHtmlOutput(`
    <!doctype html>
    <html>
      <head>
        <meta name="viewport" content="width=device-width,initial-scale=1">
        <title>Security Awareness</title>
        <style>
```

```
body { font-family: Arial, sans-serif; background:#f5f7fa; margin:0; padding:40px; display:flex; justify-content:center; }

.box { background:white; padding:28px; border-radius:10px; width:100%; max-width:720px; box-shadow:0 6px 18px rgba(0,0,0,0.08); text-align:left; }

h1 { color:#d93025; margin-top:0; }

p { color:#333; line-height:1.6; }

.actions { margin-top:20px; display:flex; gap:12px; }

.btn { background:#1a73e8; color:white; padding:12px 18px; border-radius:6px; text-decoration:none; font-weight:600; border:none; cursor:pointer; }

.btn.secondary { background:#e8eaed; color:#202124; }

</style>

</head>

<body>

<div class="box">

<h1>Phishing Simulation</h1>

<p>You have clicked a simulated phishing link as part of an authorised security exercise.</p>

<p><strong>Why this matters:</strong> Phishing is one of the top causes of breaches. Training like this helps you spot suspicious messages and protect sensitive data.</p>

<div class="actions">

<button class="btn" id="goLessons">Go to Lessons</button>

<button class="btn secondary" onclick="window.close()">Close</button>

</div>

</div>

<script>

(function(){

const microUrl = ${ JSON.stringify(microLessonsUrl) }; // injected server-side value

document.getElementById('goLessons').addEventListener('click', function(){
```

```
            if (microUrl && microUrl.length>0) {
              // preserve original email & campaignId params when redirecting
              const qp = new URLSearchParams(window.location.search);
              const email = qp.get('email') || '';
              const campaignId = qp.get('campaignId') || '';
              const final = microUrl + '?email=' + encodeURIComponent(email) +
'&campaignId=' + encodeURIComponent(campaignId);
              window.location.href = final;
            } else {
              // fallback: show a quick inline micro-tip if micro lessons URL not set
              alert('Micro-lessons not configured. Tip: Hover links before clicking and verify
sender email addresses.');
            }
          });
        })();
      </script>
    </body>
  </html>
 `);
 return html;
}
```

Notes / Required config keys (in config sheet):

MICROLESSONS_URL — set to the Micro Lessons web app URL (if you deploy MicroLessons as a separate project). If empty, the awareness page shows a fallback tip.

Templates: ensure your templates.body includes the {{redirect_url}} (or {{tracking_url}}) placeholder where you want the clickable link to appear.

---

2) Micro-lessons project — MicroLessons.gs

```
// ===== MicroLessons.gs (Standalone Project) =====

// This project should be deployed separately as a Web App

// It connects to your central spreadsheet by ID and reads micro_lessons tab

const SHEET_ID = 'PASTE_YOUR_SPREADSHEET_ID_HERE'; // <<-- REPLACE with your spreadsheet ID

const SS_MICRO = SpreadsheetApp.openById(SHEET_ID);

const MICRO_SHEET = SS_MICRO.getSheetByName('micro_lessons');

const EVENTS_SHEET = SS_MICRO.getSheetByName('events');

function doGet(e) {

  // Read micro lessons

  if (!MICRO_SHEET) return HtmlService.createHtmlOutput('Micro lessons sheet not found.');

  const rows = MICRO_SHEET.getDataRange().getValues();

  if (rows.length <= 1) {

    return HtmlService.createHtmlOutput('No micro lessons available.');

  }

  const headers = rows[0].map(h => String(h).trim());

  const dataRows = rows.slice(1).map(r => {

    return {

      id: r[0],

      lesson_title: r[1] || '',

      message: r[2] || '',
```

```
    video_url: r[3] || ''
  };
});
// Optionally log that the lessons page was served (not required).
try {
  const email = e.parameter.email || '';
  const campaignId = e.parameter.campaignId || '';
  if (EVENTS_SHEET) {
    EVENTS_SHEET.appendRow([Utilities.getUuid(), new Date(), email, campaignId, '', 'lessons_viewed', '', '', '', 'served lessons page']);
  }
} catch (err) {
  // ignore logging errors
}
// Build HTML UI showing lessons as cards and modal viewer
const html = HtmlService.createHtmlOutput(`
  <!doctype html>
  <html>
    <head>
      <meta name="viewport" content="width=device-width,initial-scale=1">
      <title>Micro Lessons</title>
      <style>
        body { font-family: Arial, sans-serif; background:#f1f3f4; margin:0; padding:20px; }
        header { background:#1a73e8; color:white; padding:18px; text-align:center; font-size:20px; }
        .container { max-width:960px; margin:20px auto; }
```

```
        .cards { display:flex; flex-wrap:wrap; gap:16px; justify-content:center; }

        .card { background:white; width:300px; padding:18px; border-radius:8px; box-shadow:0 6px 14px rgba(0,0,0,0.08); cursor:pointer; }

        .card h3 { margin:0 0 8px 0; color:#1a73e8; }

        .card p { color:#333; font-size:14px; line-height:1.4; }

        .modal { display:none; position:fixed; inset:0; background:rgba(0,0,0,0.6); justify-content:center; align-items:center; }

        .modal-content { background:white; width:90%; max-width:800px; padding:20px; border-radius:8px; }

        .close { float:right; cursor:pointer; font-size:20px; color:#777; }

        iframe { width:100%; height:420px; border-radius:6px; border:0; }

        .backlink { display:inline-block; margin-bottom:18px; text-decoration:none; color:#1a73e8; font-weight:600; }

      </style>

    </head>

    <body>

      <header>Cybersecurity Micro Lessons</header>

      <div class="container">

        <a class="backlink" href="${escapeHtml(getReturnUrl(e))}">← Back to Awareness Page</a>

        <p>Click a lesson to open it:</p>

        <div class="cards">

          ${dataRows.map((d, idx) => `

            <div class="card" onclick="openLesson(${idx})">

              <h3>${escapeHtml(d.lesson_title)}</h3>

              <p>${escapeHtml((d.message || '').substring(0,160))}${(d.message || '').length>160 ? '...' : ''}</p>

            </div>
```

```
        `).join('')}
      </div>
    </div>
    <div class="modal" id="modal">
      <div class="modal-content">
        <span class="close" onclick="closeModal()">×</span>
        <h2 id="modalTitle"></h2>
        <p id="modalMessage"></p>
        <div id="modalVideoWrap"></div>
      </div>
    </div>
    <script>
      const lessons = ${JSON.stringify(dataRows)};
      function openLesson(i) {
        const l = lessons[i];
        document.getElementById('modalTitle').innerText = l.lesson_title;
        document.getElementById('modalMessage').innerText = l.message;
        const wrap = document.getElementById('modalVideoWrap');
        wrap.innerHTML = l.video_url ? '<iframe src="'+ l.video_url +'"
allowfullscreen></iframe>' : '';
        document.getElementById('modal').style.display = 'flex';
)
      }
      function closeModal() {
        document.getElementById('modalVideoWrap').innerHTML = '';
        document.getElementById('modal').style.display = 'none';
```

```
        }
      </script>
    </body>
  </html>
 `);
 return html.setXFrameOptionsMode(HtmlService.XFrameOptionsMode.ALLOWALL);
}
// Helper server-side function to build a return link back to awareness page
function getReturnUrl(e) {
 // If the caller passed a return URL param, use it; otherwise empty #
 try {
  const params = e.parameter || {};
  const returnUrl = params.returnUrl || '';
  return returnUrl || '#';
 } catch (err) {
  return '#';
 }
}
// simple HTML escape for server-side injection
function escapeHtml(str) {
 if (!str) return '';
 return
String(str).replace(/&/g,'&amp;').replace(/</g,'&lt;').replace(/>/g,'&gt;').replace(/"/g,'&quot;');
}
```

3) templates.body example (how to place redirect link)

```
<p>Dear {{name}},</p>
```

We detected unusual activity on your account. Please verify your account now by clicking the link below:

Verify your account

Regards,
Security Team

When sendCampaignById runs it will replace {{redirect_url}} with a tracking URL built from WEBAPP_URL.

---

4) Required sheet tabs & header rows (exact text)

Create these tabs and headers (first row):

config

key | value

(example rows: WEBAPP_URL, MICROLESSONS_URL, SENDER_NAME, SENDER_EMAIL)

campaigns

id | campaign_name | start_date | end_date | status | template_id | total_sent | total_clicked | total_reported

templates

id | template_name | subject | body | redirect_url

(body should be HTML; include {{redirect_url}})

recipients

id | name | email | department | status | last_clicked | last_reported

(put status as active for test recipients)

events

id | timestamp | recipient_id_or_email | campaign_id | template_id | action | device | browser | location | extra

micro_lessons (for the MicroLessons project)

id | lesson_title | message | video_url

(video_url may be an embeddable YouTube https://www.youtube.com/embed/... or blank)

---

5) Deployment checklist (step-by-step)

A. Main project (Code.gs)

1. Paste Code.gs into the Apps Script editor bound to your PhishPortal_Data sheet.

2. Save.

3. Deploy the project as Web App: Deploy → New deployment → Web app.

Execute as: Me (your account)

Who has access: Anyone (or Anyone within <your-domain> for internal testing)

4. Copy the Web App URL.

5. Paste the Web App URL into config sheet value for key WEBAPP_URL.

6. In config set SENDER_NAME and SENDER_EMAIL if desired.

7. Run testSend() once from the Editor to authorize scopes and to test sending to a small batch.

B. MicroLessons project

1. Create a new Apps Script project (not bound).

2. Paste MicroLessons.gs. Replace SHEET_ID. Save.

3. Deploy as Web App (Execute as: Me; Access: Anyone).

4. Copy MicroLessons web app URL. Paste into main project's config sheet value for key MICROLESSONS_URL.

C. Send a test campaign

1. Ensure campaigns has a campaign row with id = '1' and template_id = '1'.

2. Ensure templates has id = '1' with body containing {{redirect_url}}.

3. Ensure recipients has at least one active recipient with email you control.

4. In Apps Script editor (main project) run sendCampaignById('1', 1) via a helper testSend() or create testSend() to call it.

D. Test the flow

Check Gmail Sent folder (sender account) to confirm email sent.

Click link in received email. It should:

1. Open main web app (awareness page), append click event to events sheet, show "Go to Lessons" button.

2. Click “Go to Lessons” button → redirect to micro lessons web app, which reads lessons from the sheet and displays them.