



**DESIGN AND DEVELOPMENT OF A WEB BASED PORTFOLIO SITE
OFFERING SOFTWARE SOLUTIONS**

BY

AFENFIA OYINDEINMUNO MITCHEL

ENG2006279

DEPARTMENT OF COMPUTER ENGINEERING

FACULTY OF ENGINEERING

UNIVERSITY OF BENIN

BENIN CITY

EDO STATE

OCTOBER, 2025

**DESIGN AND DEVELOPMENT OF A WEB BASED PORTFOLIO SITE
OFFRERING SOFTWARE SOLUTIONS**

BY

**AFENFIA OYINDEINMUNO MITCHEL
ENG2006279**

**PROJECT SUBMITTED TO THE DEPARTMENT OF COMPUTER
ENGINEERING**

FACULTY OF ENGINEERING

UNIVERSITY OF BENIN, BENIN CITY

**IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD
OF BACHELOR OF ENGINEERING (B.ENG) DEGREE IN COMPUTER
ENGINEERING**

OCTOBER, 2025

CERTIFICATION

I hereby certify that this project **DESIGN AND DEVELOPMENT OF A WEB BASED PORTFOLIO SITE OFFRERING SOFTWARE SOLUTIONS** for the award of B.Eng. conducted and duly presented by Afenia Oyindeinmuno Mitchel (ENG2006279) of The Department of Computer Engineering, Faculty of Engineering, University of Benin, Benin City and is hereby certified.

.....

Prof S.T. Apeh
(Project Supervisor)

.....

Date

.....

Engr. Dr. Isi Edeoghon
(Head of Department)

.....

Date

DEDICATION

This project is dedicated to God Almighty, whose grace, wisdom, and strength have guided me throughout this academic journey. His unfailing presence has been my source of inspiration and perseverance.

I also dedicate this work to my beloved family, whose unwavering love, support, and encouragement have been instrumental in my pursuit of excellence. Your sacrifices and belief in me have continuously fueled my determination to achieve my goals.

This project is a testament to the dedication, patience, and guidance of those who have stood by me, and I remain forever grateful.

ACKNOWLEDGEMENT

I sincerely express my deepest gratitude to God Almighty, the Wellspring of Knowledge, for His divine guidance, favor, and wisdom that have illuminated my path throughout this academic journey. His grace and strength have been my anchor, making this achievement possible.

I extend my heartfelt appreciation to my supervisor, Prof. Simon T. Apeh for his unwavering support, mentorship, and invaluable guidance throughout this project. His expertise, patience, and constructive feedback have played a crucial role in shaping this research.

The Computer Engineering Department staff as well as lecturers deeply deserve my heartfelt thanks for both delivering outstanding knowledge and establishing an excellent learning space that deepened my academic achievements. Teaching and mentoring at their core have directly developed both my technical capabilities and research competencies.

The success I achieved exists because of my parents, Mr. and Mrs. Afenia and family members who selflessly gave up everything to support me with their endless love. Their unwavering support functions as my strongest driving force which propels me to push towards achieving top-class results.

I deeply acknowledge my friends and study group family and their joint dedication to academic success which enriched my academic path with both positive memories and productive results. Their friendship has provided me with essential aid that reduces the difficulty of challenging times.

I express boundless gratitude to every person who played any role in making this work successful. This work demonstrates the invaluable support supplied by everybody who accompanied me on this learning path.

ABSTRACT

This project presents the design and development of a web-based portfolio site offering software solutions. The system was conceived to provide a professional online platform for individuals or organizations to showcase their technical expertise, previous projects, and range of offered services. The growing dependence on digital platforms for business visibility has made online portfolios a vital tool for branding and client engagement. The project was implemented using a combination of front-end technologies — HTML for content structure, CSS for styling, and JavaScript for interactivity — while Django, a Python-based web framework, was employed for back-end development and server-side processing. PostgreSQL was used as the database management system to ensure reliable data storage and retrieval.

The website features multiple modules, including a landing page, services page, about page, contact form, and an administrative dashboard for managing content. Additional functionalities such as user authentication, task management for the development team, and an integrated payment system were incorporated to extend the platform's usability for both clients and administrators. The system was tested manually to ensure that all modules functioned according to specification, and results confirmed efficient performance, responsiveness, and ease of navigation.

Overall, the project successfully demonstrates the application of modern web technologies in developing an interactive and dynamic portfolio website capable of promoting professional services online. It provides a scalable foundation that can be extended in the future to include advanced content management, analytics integration, and broader service offerings.

Keywords: Portfolio Website, Django, Web Development, Software Solutions, Database Management, HTML, CSS, JavaScript, PostgreSQL.

TABLE OF CONTENTS

CHAPTER ONE: INTRODUCTION

1.1 Background of the Study	1
1.2 Problem Statement	2
1.3 Aim and Objectives	3
1.3.1 Aim	3
1.3.2 Objectives	4
1.4 Scope of the Study	4
1.5 Relevance of the Study	5

CHAPTER TWO: LITERATURE REVIEW

2.1 Introduction	7
2.1.1 Definition of Portfolio Website	7
2.1.2 Importance and Benefits	8
2.1.3 Key Features	8
2.1.4 Design and User Experience	9
2.1.5 Branding	9
2.1.6 Credibility and Trust	10
2.1.7 Accessibility and Inclusivity	10
2.1.8 Searchability and SEO	10
2.2 Understanding the Portfolio Website Research	11
2.3 Challenges Associated with Portfolio Website Research	11
2.4 Review of Related Works	12
2.5 Research Gaps	13
2.6 Review and Comparison of Portfolio Website Services	14
2.7 Review of Tools and Technologies to be Deployed	16

CHAPTER THREE: METHODOLOGY

3.1 System Development Methodology (Agile Model)	18
3.2 System Requirements	18

3.2.1 Functional Requirements	19
3.2.2 Non-functional Requirements	19
3.3 System Design Architecture	20
3.4 Choice of Tools and Programming Languages	21
3.5 Use Case Diagram	23

CHAPTER FOUR: RESULTS AND DISCUSSION

4.1 Platform Overview	26
4.2 Key Features of the Developed Platform	30
4.3 Discussions	38
4.4 Project Timeline (Gantt Chart)	40

CHAPTER FIVE: CONCLUSION

5.1 Summary of the Research	43
5.2 Expectation of the Research Outcome	43
5.3 Challenges Faced	44
5.4 Expected Contribution to Knowledge	46
5.5 Achievements of the Study	47
References	49

LIST OF ACRONYMS

API – Application Programming Interface
CSS – Cascading Style Sheets
DBMS – Database Management System
HTML – Hypertext Markup Language
HTTP – Hypertext Transfer Protocol
HTTPS – Hypertext Transfer Protocol Secure
JS – JavaScript
MVC – Model–View–Controller
SQL – Structured Query Language
UI – User Interface
UX – User Experience
URL – Uniform Resource Locator
WWW – World Wide Web
CRUD – Create, Read, Update, Delete

LIST OF TABLES

Table 2.1: Comparison of Portfolio Website Services	15
Table 3.1: Materials for Development	21

LIST OF FIGURES

Figure 3.1: Use Case Diagram	25
Figure 4.1: Home Page	31
Figure 4.2: About Us Page	32
Figure 4.3: Services Page	32
Figure 4.4: Join Us Page	33
Figure 4.5: Complaints Page	34
Figure 4.6: Payment Page	35
Figure 4.7: Admin Page	36
Figure 4.8: Gantt Chart	42

CHAPTER ONE

INTRODUCTION

1.1 BACKGROUND OF THE STUDY

In the modern digital economy, businesses increasingly rely on web presence to attract customers and build credibility. A portfolio website enables a software solutions provider to showcase its services, projects, and expertise to potential clients, serving as a central hub for brand communication and customer engagement (Nwali and Ntegeeh, 2022). Research indicates that strong digital marketing and online presence can significantly improve the performance and visibility of small and medium enterprises (SMEs) in Nigeria (Ajede *et al.*, 2025). For example, Ajede *et al.* (2025) report that Nigerian SMEs which embrace online marketing are recommended to increase product awareness and business visibility through channels like social media and email. Similarly, Nwali and Ntegeeh (2022) found that social media marketing enhances organizational brand awareness and performance for Nigerian SMEs. A professional portfolio site complements social media efforts by providing a polished, centralized platform for information, thereby reinforcing trust and brand identity.

Despite rising connectivity in Nigeria, digital adoption by businesses remains uneven. As of early 2023, Nigeria had about 122.5 million internet users (roughly 55% of the population). Moreover, mobile connectivity is extremely high – 193.9 million cellular connections (87.7% of the population) – indicating that most Nigerians have access to the internet via mobile devices. However, Research ICT Africa (2024) reports that only 13% of microenterprises in Nigeria use the internet for business activities. This suggests a gap between availability of connectivity and its effective use for commerce. When small businesses do go online, they benefit from reduced communication costs and faster information sharing. These findings imply that a well-designed web portfolio could help Nigerian software firms reach wider audiences and operate more efficiently.

Modern user expectations make website quality crucial. Users form impressions of a site in mere fractions of a second, with visual design greatly influencing their judgment (Tractinsky *et al.*, 2000). For instance, research indicates that 94% of users' first impressions are design-related, and poor interface design can trigger rapid rejection

of a site. Thus, a contemporary portfolio site must feature intuitive navigation, appealing visuals, and mobile-responsive layouts to retain users. The project will emphasize modern UI/UX practices (e.g. consistent typography, ample white space, and clear calls to action) so that visitors quickly perceive professionalism and trustworthiness.

On the technical side, widely adopted web development tools support this endeavor. Frontend interfaces are typically built with HTML5, CSS3, and JavaScript for interactivity. For the backend, the Python-based Django framework is popular for full-stack development. A 2021 survey of Django developers found that PostgreSQL is the leading database choice (77% preference), and most users combine Django with HTML/CSS (67%) and JavaScript (75%). These technologies will be used to implement the portfolio site, providing a robust foundation for dynamic content (e.g. project entries, contact forms) and secure data management. By employing such modern tools, the project aligns with current industry practice and ensures scalability of the solution.

Locally, Nigeria's digital economy continues to grow. Guanah and Bebenimibo (2025) conclude that digital technology has the potential to transform Nigeria's economy by creating new business opportunities and enhancing access to markets. In this context, a professional web portfolio can help a software company tap into that potential. By delivering a polished online presence, this project supports the broader trend of digital transformation in Nigerian business (Guanah and Bebenimibo, 2025). Thus, the background of this study underscores the importance of combining effective web design and development to advance the competitiveness of software solution providers in Nigeria.

1.2 PROBLEM STATEMENT

In Nigeria, many software development and consulting firms lack a professional, user-friendly online portfolio. The absence of such a platform constrains their ability to reach potential clients and showcase their work. Without an interactive website, prospective customers may have difficulty finding information about the company's services and past projects, reducing the firm's visibility. Studies show that consumers often consider a company's online presence when making buying decisions (Nwali and Ntegeeh, 2022); thus, a weak or nonexistent web presence can lead to missed

opportunities. Similarly, Ajede *et al.* (2025) note that SMEs with limited online marketing find it nearly impossible to compete favorably and improve performance (Ajede *et al.*, 2025).

Furthermore, existing websites may suffer from poor usability and design. If a site is visually outdated or difficult to navigate, users are likely to form negative impressions and leave quickly (Tractinsky *et al.*, 2000). In Nigeria's context, many small businesses lack the resources or expertise to implement modern UI/UX standards. Microenterprises often cite lack of knowledge and high costs as barriers to adopting digital tools. As Research ICT Africa (2024) reports, obstacles such as unaffordable devices and low digital literacy prevent the majority of informal businesses from leveraging online channels.

The result is that software firms and freelancers in Nigeria frequently rely on word-of-mouth or rudimentary social media pages, instead of a cohesive web platform. This limits customer trust and undermines brand identity. The problem this project addresses is the design and development of a comprehensive web-based portfolio site that remedies these shortcomings. The new site must meet contemporary user expectations (e.g. responsive layout, intuitive interface) and use appropriate technologies to ensure maintainability and scalability. By solving these problems, the portfolio site will help the company improve its brand visibility, client engagement, and competitive positioning in the digital landscape (Nwali and Ntegeeh, 2022).

1.3 AIM AND OBJECTIVES

1.3.1 AIM

The aim of this project is to design and develop a web-based portfolio website for a software solutions provider, enabling effective presentation of the company's services, past projects, and contact information in a professional and accessible manner. This site should enhance the company's brand image, facilitate client acquisition, and demonstrate the application of modern web development technologies.

1.3.2 OBJECTIVES

Literature and market review: Investigate existing portfolio websites and identify best practices in UI/UX design and web development for technology service providers.

1. Requirement analysis: Elicit and document the functional and non-functional requirements of the portfolio site by considering user needs (e.g. ease of navigation, content organization) and business goals.
2. System design: Create the site architecture and design the user interface wireframes, ensuring consistency, responsiveness, and alignment with modern design principles.
3. Implementation: Develop the portfolio site using HTML5, CSS3, and JavaScript for the frontend, and Python with the Django framework and PostgreSQL for the backend. Incorporate features such as project showcases, an about section, and a contact form.
4. Testing and evaluation: Test the implemented site for functionality, usability, and performance. Collect user feedback to refine the design and ensure the site meets quality standards.
5. Deployment: Deploy the completed website to a suitable hosting environment and ensure it is accessible on various devices.

These objectives will guide the project through systematic stages of analysis, design, construction, and validation, resulting in a functional web portfolio site.

1.4 SCOPE OF THE STUDY

This project focuses on the design and development of a single web-based portfolio site for a software solutions company. The scope includes front-end design (HTML, CSS, JavaScript) and back-end development (Python/Django, PostgreSQL database) to create the core features of the site. Specifically, the site will include an About page (company overview), a Services page (listing software solutions offered), a Portfolio/Projects section (showcasing completed projects), and a Contact page with a form for inquiries. The design will emphasize responsive layouts so that the site functions well on both desktop and mobile devices, reflecting Nigeria's high mobile connectivity rate.

The project will also cover basic administrative functionality, allowing content updates (e.g. adding new projects or blog entries) through Django's built-in admin interface. Security measures such as input validation will be implemented to protect the site's integrity. Performance optimization (e.g. efficient database queries, caching) will be considered to ensure the site loads quickly for users.

Certain features are out of scope for this study. The site will not include e-commerce or online payment processing, as it is focused on portfolio presentation rather than sales. Integration with external services (beyond basic social media links) is not required. Advanced functionalities such as user accounts for visitors, complex search, or content management beyond the implemented admin panel will be excluded. The project also will not involve marketing strategies or search engine optimization beyond fundamental metadata. The scope is deliberately bounded to allow a comprehensive prototype to be delivered within the timeframe of a final-year project.

1.5 RELEVANCE OF THE STUDY

This study is significant both academically and practically. From an educational standpoint, it provides the researcher with experience in applying theoretical knowledge of web technologies (HTML, CSS, JavaScript, Django, PostgreSQL) to build a real-world application. It also demonstrates integration of user-centered design principles within an engineering project, which is valuable in a final-year engineering curriculum.

Practically, the resulting portfolio site can serve as a model for other Nigerian SMEs and tech startups. Given that digital presence can drive business growth, this project directly addresses a need in the local industry. As Guanah and Bebenimibo (2025) conclude, digital technology can significantly transform Nigeria's economy by creating new business opportunities and expanding market access. By showcasing one way to leverage web technology, the project contributes to this digital transformation. Additionally, Ajede *et al.* (2025) recommend that more Nigerian SMEs embrace online marketing to increase visibility and customer reach; a well-designed portfolio site is a concrete step in that direction.

For businesses, an effective web portfolio enhances brand awareness. Research in Nigeria has shown that a strong online presence leads to improved brand visibility

and customer engagement (Nwali and Ntegeeh, 2022). By improving the company's web presence, this project helps meet those needs. It addresses the low rate of internet business use (13% of microenterprises) reported by Odufuwa *et al.* (2024) by providing a template that other firms might emulate. In summary, this study is relevant because it combines academic learning with practical problem-solving to support Nigeria's growing digital economy and the competitiveness of its SMEs.

CHAPTER TWO

LITERATURE REVIEW

2.1 INTRODUCTION

A portfolio website is essentially an online platform for showcasing a professional's work, skills, and achievements. Bhosle (2023) defines a portfolio website as “an online platform that showcases a professional's work, serves as a marketing tool, and provides information about the individual's skills and experience”. In other words, it functions like a digital résumé or showcase. As such, portfolio sites have become *essential* marketing instruments in the digital age. They establish a professional brand and credibility by presenting evidence of competence. For example, Bhosle (2023) notes that having a portfolio site offers benefits such as “establishing a professional brand, gaining exposure, and expanding job opportunities”. In practice, a well-designed portfolio helps creative and software professionals market their services online.

Portfolio websites typically include several key features and components. Common elements are an introduction or homepage, a detailed projects gallery, an “About Me” or profile section, and a contact form or links. According to Bhosle (2023), effective portfolio sites have clear navigation, organized layouts, engaging content, and good user experience. The use of web standards – e.g. HTML for content markup, CSS for styling, and JavaScript for interactivity – is central. These technologies build the site's structure and style: “HTML is the standard markup language for creating webpages,” while CSS and JavaScript are used for styling and interactivity respectively. Modern portfolios often use frameworks or libraries (such as Bootstrap, React, or Vue.js) to streamline development, as demonstrated by Gandla et al., who employed Bootstrap and React for the front end and Python/Flask with MySQL on the back end to create a portfolio site.

2.1.1 DEFINITION OF PORTFOLIO WEBSITE

A portfolio website is an online showcase of work and skills. It is essentially a digital collection of a person's projects, achievements, and resume that can be accessed by clients or employers. In today's digital world, such a site is often *expected* for

creative professionals. As Bhosle (2023) observes, a portfolio website “has become a necessity in today’s digital world for creative professionals to display their work, skills, and achievements”. In summary, a portfolio site is defined as an individualized website (often personal or business-branded) created to present one’s professional portfolio to the public.

2.1.2 IMPORTANCE AND BENEFITS

Portfolio websites serve multiple important functions. They act as marketing tools – effectively an online CV – and help individuals or companies gain visibility. By curating and publishing examples of past work, a portfolio establishes a professional’s *brand* and reputation. According to Bhosle (2023), such sites help “establish a professional brand, gaining exposure, and expanding job opportunities”. In practical terms, many employers now request links to a candidate’s online portfolio in job applications. A strong portfolio can directly influence hiring decisions and client trust. In addition, a dedicated website is more credible than ad-hoc social media profiles. Online presence statistics (e.g. 139.2 million active Internet users in Nigeria as of 2024) indicate that a vast audience is reachable; thus, a portfolio site is an effective way to reach potential clients and collaborators. In the context of Nigeria’s growing digital economy, SMEs and freelancers benefit from professional websites: research shows that digital marketing strategies (including websites) significantly drive revenue growth and customer acquisition for Nigerian small businesses.

2.1.3 KEY FEATURES

Key features of a good portfolio site include responsiveness, intuitive navigation, and rich content presentation. As highlighted by Bhosle (2023), essential components include a clear menu, a portfolio/projects section, about and contact information, and strong UX design. Visually, sites should use high-quality images or demos of work, consistent typography, and color schemes that reflect the brand. For example, Gandla *et al.* (cited in Bhosle, 2023) emphasize using a component-based front-end (HTML/CSS/JavaScript) with libraries like Bootstrap and React to create a clean, navigable layout. Important practical features also include social media

integration and SEO-friendly structure. Overall, ease of navigation and fast loading (often via optimization and mobile-friendly design) are key for user engagement.

2.1.4 DESIGN AND USER EXPERIENCE

Design is a critical aspect of portfolio websites. Modern portfolios must be responsive – i.e. adapt to all screen sizes (desktop, tablet, mobile) – because many clients browse on phones. Bhosle (2023) emphasizes “the significance of responsive design and user experience” in effective portfolios. A responsive, mobile-friendly layout ensures a broad audience can view the content easily. Beyond responsiveness, UI/UX design principles (clean layouts, legible text, meaningful images) are essential to communicate professionalism. Use of whitespace, consistent branding elements, and intuitive interactions all contribute to a good user experience. According to Nielsen Norman Group, visual design quality and clarity (e.g. professional graphics, no broken links) are among the primary ways a website can communicate trustworthiness to users. In summary, portfolio sites should balance aesthetic appeal with usability – for instance, a streamlined drag-and-drop editor (as offered by platforms like Wix or Weebly) can help non-technical users achieve polished designs efficiently.

2.1.5 BRANDING

Personal or corporate branding is a key purpose of a portfolio website. The site itself *becomes* part of one’s brand. All elements – logo, color palette, typography, tone of writing – should convey a coherent identity. Bhosle (2023) notes that portfolios help in “establishing a professional brand”. For example, a freelance designer’s portfolio might use a distinctive logo and color scheme to stand out, effectively reinforcing their brand “personality” (as noted by Tsykunov, 2019 in a case study of web portfolio design). The content of the portfolio – such as an “About” section – also helps craft a narrative: showing achievements and testimonials can strengthen one’s brand story. In short, portfolios function as a branding tool, by making a memorable first impression on visitors.

2.1.6 CREDIBILITY AND TRUST

A well-crafted portfolio site lends credibility to the professional. Users tend to *trust* businesses or individuals who have a polished online presence. Research in web usability shows that a website’s design quality, transparent content (clear pricing, policies), and functioning links significantly influence perceived credibility. For portfolio sites, this means maintaining up-to-date project showcases, providing clear contact information, and ensuring all pages work (no broken links or spelling errors). Indeed, a portfolio acts as an online “face” for the professional – any errors or outdated design can damage credibility. For instance, Harley (2016) emphasizes that a professional-looking site (with appropriate images and color schemes) helps establish trust. Similarly, a user testimonial or certification shown on a portfolio can boost visitor confidence. Therefore, portfolio sites should be seen as trust-building assets.

2.1.7 ACCESSIBILITY AND INCLUSIVITY

Portfolio sites should be accessible to all users, including those with disabilities. This involves practices such as using alt text for images, ensuring sufficient color contrast, and supporting keyboard navigation. A broad audience reach, including older or differently-abled users, enhances both impact and professionalism. Given Nigeria’s diversity, designers should consider multilingual support or at least clear, simple language to serve users with varying backgrounds. In addition, good performance (fast load times) is especially crucial where users may have limited bandwidth. In sum, accessibility considerations align with the overall goal of a portfolio: to present one’s work to as many people as possible.

2.1.8 SEARCHABILITY AND SEO

Portfolio websites must also be search engine optimized so that potential clients can find them. Simple steps (use of meta tags, descriptive headings, mobile optimization, and inclusion of relevant keywords) help improve visibility on search engines. With many recruiters and customers using Google to find services, SEO cannot be ignored. While no specific study was cited here, general best practices suggest that a portfolio site should be configured to appear in relevant online searches. This includes writing descriptive URLs, page titles, and using alt text on images. Overall, good SEO

helps the portfolio site reach a wider audience and further supports branding and credibility.

2.2 UNDERSTANDING THE PORTFOLIO WEBSITE RESEARCH

Research on portfolio websites covers their design, function, and impact. Much of the literature focuses on *how* to build effective portfolios (e.g. design case studies, development guides) rather than on quantitative usage statistics. For example, Bhosle (2023) and other authors provide guidelines for development and highlight technologies (HTML, CSS, JavaScript) and design principles. There is comparatively little formal research specifically about portfolio sites in the Nigerian context. However, understanding the portfolio trend requires awareness of broader tech adoption in Nigeria. Nigeria had about 139.2 million active Internet subscribers in 2024 – a very large audience for online services. This growth (despite recent declines from regulation) suggests that more Nigerians are online than ever. Additionally, studies of Nigerian SMEs show that digital marketing (including websites) has a strong impact on business growth. Together, these findings imply that creating or researching portfolio websites in Nigeria is timely. In particular, research should consider local Internet penetration (currently about 42.7% on broadband) and user behavior.

In sum, “understanding” portfolio website research means combining insights from global web design studies with local market realities. Key factors include the available technology infrastructure (Internet and mobile usage) and the digital literacy of target users. On a practical level, Nigerian professionals looking to develop a portfolio site must understand both universal design principles and how these translate to local needs. For instance, given the high mobile usage, responsive design (Bhosle, 2023) and offline readability (e.g. fast load) become critical. Likewise, portfolio research may overlap with studies of e-commerce or digital entrepreneurship in Nigeria, since all emphasize professional online presence.

2.3 CHALLENGES ASSOCIATED WITH PORTFOLIO WEBSITE RESEARCH

Several challenges arise when researching or developing portfolio websites, especially in Nigeria. Technological limitations are primary: although Nigeria has many Internet subscribers, only about 36% of the population uses the Internet regularly.

Connectivity can be uneven, with rural areas lacking broadband and frequent power outages limiting access. UNICEF (2025) reports that only 42.7% of connections are on 4G or better, and 78% of Nigerian youth lack basic digital literacy. These factors hamper both the creation and use of portfolio sites in Nigeria – developers must assume that some visitors have slow connections or basic devices.

Another challenge is the cost of access and development. High mobile data costs or computer ownership barriers can limit who can view a portfolio. Similarly, small businesses or individuals may find web development services (domain registration, hosting, design work) relatively expensive. Although platforms like GitHub Pages or free plans on Weebly/Wix mitigate costs, there is still a learning curve or premium plan required for full features.

There are also societal and adoption challenges. Many Nigerians (especially in older or conservative sectors) may not prioritize an online portfolio, relying instead on in-person networking. Low digital literacy (as noted) means some professionals are not comfortable building or maintaining a personal website. This cultural gap makes it harder to gather data on portfolio usage.

Finally, on the research side, there is a lack of local data. There are few surveys or analytics focusing on *who* uses portfolio websites in Nigeria or *how* clients find them. Academic work on portfolios is sparse, so researchers often rely on anecdotal or global industry reports. For example, the noted drop in Nigeria’s Internet subscribers (from 163.8M to 139.2M) was due to SIM registration regulations, a factor that complicates longitudinal studies of web presence. In short, infrastructure, cost, and cultural factors all pose challenges to understanding and promoting portfolio websites in the Nigerian environment.

2.4 REVIEW OF RELATED WORKS

Despite the ubiquity of portfolio sites, academic research specifically on them is limited. Bhosle (2023) provides a notable example of a design-oriented study, surveying literature and case studies to offer a comprehensive guide to portfolio web development. This paper covers technology stacks (HTML, CSS, JavaScript), responsive design, and effective features (such as navigation and project showcases). Similarly, practical

project reports exist: for instance, Etunim and Idemudia (2022) documented the “Design and Implementation of a Personal Portfolio Website” for a Nigerian context (Auchi Polytechnic). Their work identifies core modules (Profile, Projects, Contact) and notes that a portfolio can significantly improve an individual’s job prospects.

Outside such case studies, related research often comes from general website design or e-portfolio (education) literature. There are studies on portfolio management in project management, but these are not directly relevant to web portfolios. In the tech field, various articles and blogs illustrate best practices or examples of portfolios, but they rarely present empirical findings. For example, industry reviews compare site builder platforms (Wix, Squarespace, etc.) for portfolios, but academic comparisons are rare. In summary, existing work tends to be either anecdotal (design showcases) or high-level (digital marketing in Nigeria). The few formally published works on portfolio websites emphasize the importance of responsive design and UX, as well as the need for personal branding and clear information. However, these studies often focus on technical and design aspects rather than user perception or business impact.

2.5 RESEARCH GAPS

The review above highlights several gaps. There is a geographic gap: almost no scholarly work examines portfolio websites specifically in Nigeria’s business environment. Most design guides (e.g. Bhosle, 2023) are from other countries, and Nigerian studies (like Etunim & Idemudia, 2022) are few and often unpublished theses. Thus, one gap is localized research – for example, user needs or preferences of Nigerian clients looking at portfolios.

Another gap is in outcome evaluation. We found little data on how portfolio websites actually affect job outcomes or sales for African freelancers and SMEs. Do Nigerian employers prefer candidates with portfolios? How much do online portfolios contribute to winning projects in Nigeria’s software industry? These questions lack empirical answers.

There is also a gap in comparative analysis of tools and services. Many platform comparisons exist globally, but none focus on factors like local affordability or accessibility. For instance, a global review might rate Wix highly (Simpson, 2025) and

note Squarespace’s elegance (Ryan, 2025), but it is unclear how these platforms perform in Nigeria (e.g. regional support, local pricing). Similarly, although frameworks like React or Laravel are popular, we have not found research on which tech stacks are most effective for portfolios in resource-constrained settings.

Finally, as alluded in 2.3, there is a lack of data on internet users’ behavior in Nigeria related to portfolio websites. Given the substantial challenges (low internet penetration, power issues), future research should investigate how these factors influence portfolio usage. In short, more empirical, locally-grounded studies are needed to fill these gaps.

2.6 REVIEW AND COMPARISON OF PORTFOLIO WEBSITE SERVICES

A number of online services and website builders offer portfolio sites. This review considers five popular platforms: Weebly, Wix, Squarespace, Adobe Portfolio, and GitHub Pages. Table 2.1 summarizes how they compare in overall rating, accessibility (ease of use), affordability (cost of plans), user experience, and customization capabilities.

Website	Wix	Squarespace	Weebly	Adobe Portfolio	GitHub Pages
Overall Rating	4.8 / 5	4.5 / 5	~4.0 / 5	4.0 / 5	~4.0 / 5
Link	wix.com	squarespace.com	weebly.com	portfolio.adobe.com	pages.github.com
Accessibility	High (WYSIWYG editor)	High (drag-and-drop)	High (simple builder)	Medium (simple but requires CC)	Low (requires coding)
Affordability	Moderate (free plan, \$\$ plans)	Lower (no free plan, \$\$\$)	High (includes free, \$)	Included with Adobe CC	Free (no cost)
User Experience	Excellent (intuitive UI)	Excellent (professional design)	Good (basic, dated templates)	Good (drag-drop, streamlined)	Good (fast hosting)
Customization	Very High (2000+ templates)	High (180+ templates)	Moderate (fewer themes)	Limited (few template options)	Very High (code-level control)

Table 2.1: Comparison of Portfolio Website Services

Wix, for example, consistently earns top ratings. WebsiteBuilderExpert’s 2025 analysis gives Wix an “Outstanding” 4.8/5 score, praising its vast template library and easy editor. Squarespace is rated around 4.5/5, noted for its high-quality design templates and overall polish. Weebly scores lower: many reviewers note that its feature set has not advanced much since its acquisition by Square, making its plans cheap but its functionality limited. Adobe Portfolio (free with a Creative Cloud subscription) is valued for seamless integration with Adobe apps, though it offers fewer customization features than others. GitHub Pages (a static-hosting service) is effectively free and customizable for developers, but it is less accessible to non-technical users. In short, each platform has trade-offs in cost, user-friendliness, and flexibility. The table above reflects these trade-offs (e.g. Wix’s generous free tier vs. Squarespace’s lack of a free plan, or GitHub’s free cost vs. steep learning curve). These platform comparisons suggest that accessibility (ease of use) is generally high for drag-and-drop builders (Wix, Weebly, Squarespace) and lower for developer tools (GitHub). Affordability varies: GitHub is free, Weebly/Wix have free basic plans, whereas Squarespace requires a paid subscription. Expert reviews validate these impressions.

2.7 REVIEW OF TOOLS AND TECHNOLOGIES TO BE DEPLOYED

Building the proposed web-based portfolio will involve a combination of front-end and back-end tools. For the front end, standard web technologies will be used: HTML5 for markup, CSS3 for styling, and JavaScript for interactivity. As noted by Bhosle (2023), HTML/CSS/JS are the “primary technologies used to build portfolio websites”. Frameworks like Bootstrap (for responsive design) will be used to accelerate development and improve maintainability. In fact, Bhosle’s review cites a case where Bootstrap was used for a modern portfolio interface.

For the back end, Python with Django is used and a PostgreSQL database to store any dynamic content. For development, code editors such as Visual Studio Code is employed. Deployment may utilize cloud platforms such as Render.com. We will ensure the site is mobile-responsive (as per Bhosle, 2023) and accessible. Security best

practices (e.g. HTTPS, input validation) will be followed to maintain credibility. In summary, the technology stack will combine HTML/CSS/JS (with possible frameworks), a suitable back-end language (Python/Node), a database (if needed), and modern development tools. This aligns with the approaches found in the literature, which emphasize these core technologies for effective portfolio sites.

CHAPTER THREE

METHODOLOGY

This chapter outlines the design and development process of the web-based portfolio site that offers software solutions. This chapter describes the methodology adopted for development, system requirements, architectural design, data flow, chosen tools, and the implementation plan. Emphasis is placed on the rationale for each choice and the systematic approach taken in building the portfolio system. The structured approach ensures that the system meets its objectives effectively.

3.1 SYSTEM DEVELOPMENT METHODOLOGY (AGILE MODEL)

The system development follows the Agile methodology, an iterative and incremental approach well-suited for evolving requirements. Agile is characterized by short cycles called sprints, frequent reassessment of tasks, and close collaboration with stakeholders. In this project, agile practices allow continuous feedback and adaptation as features of the portfolio site are refined over time.

Each sprint begins with planning, where requirements are broken into user stories and tasks. Development is carried out in iterations, each producing a usable product increment. Daily stand-ups (or personal progress checks) track progress and identify obstacles. Sprints typically last one to two weeks, focusing on specific aspects such as user interface design, back-end functionality, or database integration.

By adopting Agile, the project can incorporate changes based on feedback without derailing the schedule. This approach ensures priorities are regularly reviewed, enabling the development to stay aligned with the project objectives. Overall, the agile model improves flexibility, allows early testing of features, and reduces the risk of misaligned deliverables.

3.2 SYSTEM REQUIREMENTS

The system requirements define what the portfolio site must do and how it must perform. They ensure that the system meets the needs of its users and stakeholders. These requirements are categorized into functional and non-functional requirements.

3.2.1 Functional Requirements

Functional requirements specify the behavior and functions of the system. For the web-based portfolio site, the functional requirements include:

- **User Interface Navigation:** The system shall present a clear navigation menu allowing users to access sections such as Home, About, Portfolio, Services, and Contact.
- **Portfolio Showcase:** The system shall display software projects, each with title, description, technologies used, and related media (images).
- **Search and Filtering:** The system shall provide search and filtering of portfolio items by keywords, categories, or tags.
- **User Authentication:** The system shall provide secure login for the administrator to manage content. Visitors do not require login for browsing.
- **Content Management:** The system shall allow the administrator to add, update, or delete content through an admin interface.
- **Contact Form:** The system shall provide a form for visitors to send inquiries or messages to the site owner.
- **Email Notifications:** The system shall send an email to the owner when a new contact message is submitted.
- **Responsive Design:** The system shall adapt its layout to different device sizes (mobile, tablet, desktop).
- **Multimedia Support:** The system shall display images associated with portfolio items.

These functional requirements ensure that the portfolio site provides a complete user experience for both visitors and the administrator.

3.2.2 Non-functional Requirements

Non-functional requirements define the system's operational qualities and constraints. For the portfolio site, the non-functional requirements include:

- **Performance and Scalability:** The system shall load pages quickly and handle multiple users effectively, with an architecture that supports future growth.

- Usability: The interface shall be user-friendly, with clear navigation and logically organized content, following best practices for readability and accessibility.
- Reliability: The system shall be reliable and available to users, handling errors gracefully to ensure minimal downtime.
- Security: Sensitive data shall be protected. The system shall use HTTPS, secure password storage, and validate input to prevent attacks like SQL injection or XSS.
- Maintainability: The code shall follow standard conventions and be well documented to facilitate maintenance and future enhancements.
- Compatibility: The site shall work on major browsers (Chrome, Firefox, Safari, Edge) and degrade gracefully on older versions.
- Accessibility: The site shall follow web accessibility guidelines to ensure it can be used by people with disabilities (e.g., providing alt text for images and clear labels for forms).

These non-functional requirements ensure that the system is robust, secure, and user-friendly.

3.3 SYSTEM DESIGN ARCHITECTURE

The system architecture describes the high-level structure of the portfolio site. The web-based portfolio follows a multi-tier architecture with three main layers: the presentation layer (client side), the application layer (server side), and the data layer (database).

The presentation layer is implemented using HTML, CSS, and JavaScript, which render the user interface in the client's browser. HTML provides the structure of each web page, CSS defines visual styles, and JavaScript adds interactivity (such as dynamic menus or form validation). This layer ensures the portfolio site is responsive and visually engaging on various devices.

The application layer is built with the Django framework (a Python-based web framework). Django follows the Model-View-Template (MVT) architectural pattern, which separates data, logic, and presentation. In this layer, *Models* represent the data

structures (for example, portfolio projects and contact messages), *Views* handle HTTP requests and business logic, and *Templates* define the layout of the web pages. Django's ORM handles communication between the application and the database.

The data layer uses PostgreSQL as the relational database management system. Project details, user credentials, and contact messages are stored in database tables. When data is requested or updated, Django's ORM translates these actions into SQL commands on PostgreSQL to maintain the data stores.

In a typical user interaction, the user's browser sends an HTTP request to the Django application on the server. The application processes the request in the appropriate view, accesses the database as needed, and renders an HTML template in response. That response is sent back to the browser, where client-side scripts may run for additional interactivity.

This separation of concerns (presentation, logic, data) makes the system modular and easier to maintain. By using Django, the application benefits from a ready-made admin interface for content management, as well as built-in security and scalability. Overall, this architecture supports a maintainable and efficient portfolio site.

3.4 CHOICE OF TOOLS AND PROGRAMMING LANGUAGES

The choice of development tools and languages is critical to the success of the portfolio site. The following tools and technologies are selected based on their suitability:

S/N	MATERIAL/TOOL	USE
1.	HTML	Building the structure and layout of the platform's web pages
2.	CSS	Styling the platform and ensuring a consistent

		responsive design
3.	JavaScript	Adding interactivity; client-side validation, and dynamic content rendering
4.	Django (Python Framework)	Backend framework for handling server-side logic, authentication and data management
5.	PostgreSQL	Database system for storing user records, livestock listings, and transaction data
6.	Render	Cloud hosting platform used for deploying and maintaining the live version of the web application

Table 3.1: Materials for Development

- **HTML and CSS:** HTML (HyperText Markup Language) and CSS (Cascading Style Sheets) are the foundational languages for structuring and styling web pages. HTML defines the elements on each page (headings, paragraphs, images, links), while CSS specifies visual design (colors, layout, fonts). Together, they enable the creation of responsive, well-formatted web pages that appear consistently across browsers. They are chosen for their ubiquity and broad support in web development.
- **JavaScript:** JavaScript is used to add interactivity on the client side. It can manipulate the content dynamically, validate form data before submission, and enhance user experience with features like dynamic galleries or menus.

JavaScript is widely supported by all browsers, making it an essential tool for front-end functionality.

- **Django (Python Framework):** Django is a high-level open-source web framework written in Python. It follows the MVC design pattern (known as MVT in Django) and offers a powerful set of tools for rapid web development. Django is chosen for its built-in ORM for database interactions, its ready-made administrative interface, and robust security features. The readability of Python and the framework's structure contribute to maintainable and robust back-end code.
- **PostgreSQL:** PostgreSQL is the chosen relational database management system. It is open-source, reliable, and known for its performance and ability to handle complex queries. PostgreSQL supports large datasets and ensures data integrity, which is important for storing detailed project information and user data. It integrates seamlessly with Django's ORM, facilitating easy database setup and migrations.

Each of these tools contributes essential capabilities. The front-end technologies deliver a rich user interface, while Django and PostgreSQL provide a robust back-end for data management. This technology stack meets the functional and non-functional requirements of the portfolio site.

3.5 USE CASE DIAGRAM

The use case diagram provides a visual representation of the interactions between the different users (actors) and the system. It illustrates how the various users — such as the administrator and general users — interact with different functional components of the web-based portfolio system.

The diagram serves as a blueprint for understanding the system's behavioral structure and defining its core functional requirements. It identifies the primary actors involved, the actions they can perform, and how these actions relate to different modules of the platform, such as user authentication, contact form submission, complaint management, payment processing, and administrative operations.

By presenting the system's functionality in this visual format, the use case diagram ensures clarity in defining user roles and system boundaries, thereby guiding both the design and implementation phases of the project.

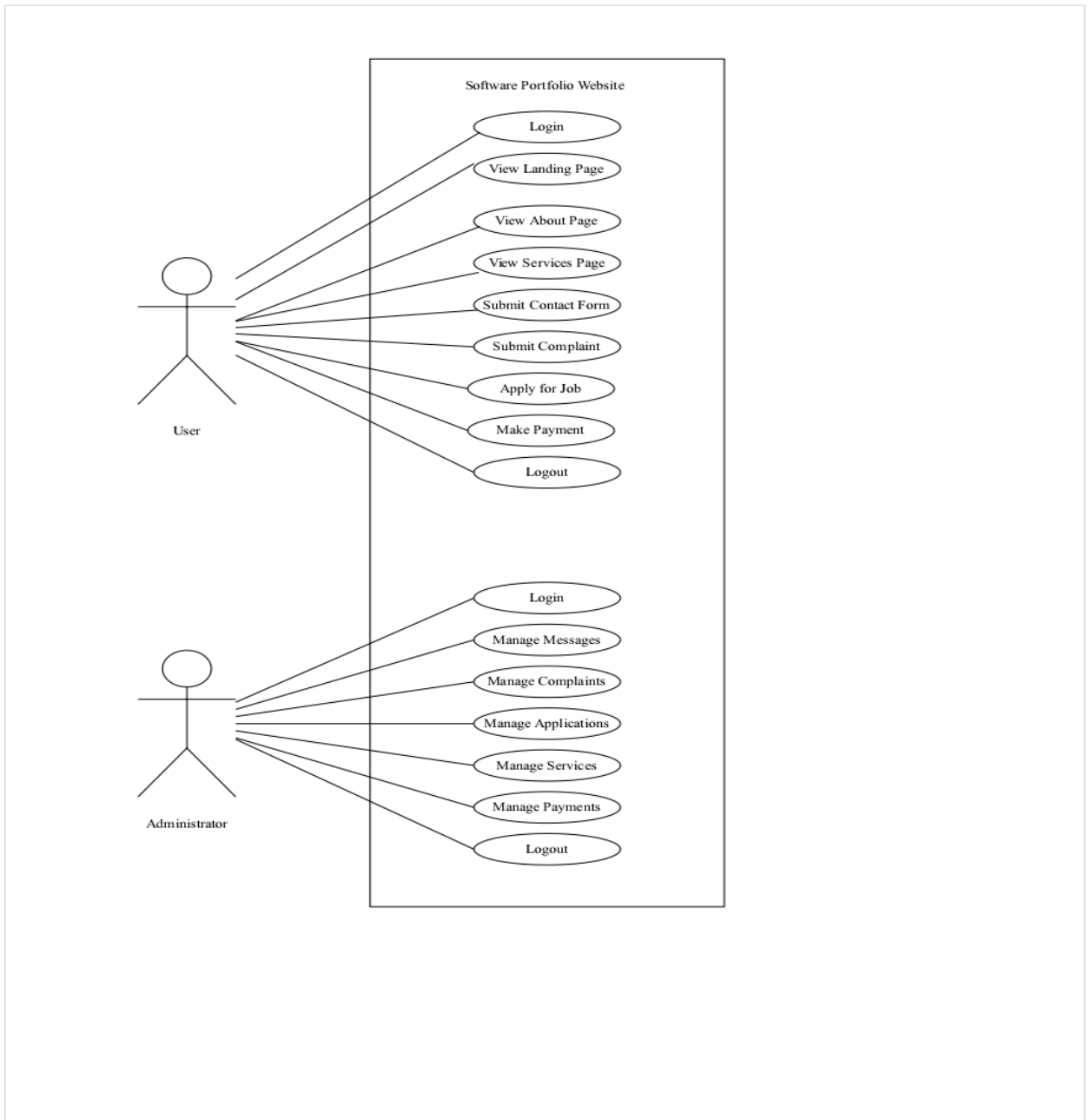


FIG 3.1: Use Case Diagram

CHAPTER FOUR

RESULTS AND DISCUSSION

This chapter presents and discusses the results obtained from the design and development of the web-based portfolio site offering software solutions. It focuses on the outcomes of the implementation phase, the system's operational performance, and how these results align with the project's stated objectives. The results are derived from the successful execution of the design and methodology described in the preceding chapters.

The chapter begins by providing an overview of the implemented platform, highlighting the main system components and user interfaces. It then outlines the key features that were developed and the outcomes of each functional module, including the landing page, service listing, complaint management, payment module, and administrative dashboard. This is followed by a discussion section that interprets these results in relation to usability, performance, security, and the overall project goals.

The essence of this chapter is to demonstrate that the system meets the functional, non-functional, and technical requirements established during the design phase. Each implemented component is evaluated based on its efficiency, accuracy, and contribution to the overall user experience. The discussion also covers the challenges encountered during system testing and the strategies adopted to resolve them.

By the end of this chapter, the reader gains a comprehensive understanding of how the system operates, the extent to which the project objectives were achieved, and the lessons learned during the implementation and testing stages. The findings presented here serve as a foundation for the conclusions and recommendations discussed in Chapter Five.

4.1 PLATFORM OVERVIEW

The developed system is a fully functional web-based portfolio platform that successfully meets the objectives outlined in Chapter One. It provides an interactive

digital presence for a software solutions company and enables seamless communication between clients, administrators, and potential employees. The platform integrates several modules that operate together to deliver a dynamic, responsive, and user-friendly experience.

The final system is structured into two main user environments:

- The Public Interface (Client View), which allows users to access general information, browse services, submit forms, and make payments; and
- The Administrative Interface (Dashboard View), which enables authorized personnel to manage and monitor submitted data in real time.

This structure ensures a clear separation between public user activities and administrative operations, enhancing both functionality and data security.

- **System Objectives Achieved**

The platform successfully fulfills its core objectives, which include:

1. Establishing a professional online platform that showcases the company's services, mission, and portfolio.
2. Providing interactive features for inquiries, job applications, and complaints.
3. Implementing a secure authentication system for administrators.
4. Integrating a structured database for storing and managing user data efficiently.
5. Delivering a responsive, accessible, and scalable website compatible with multiple devices.

All these objectives were achieved through the combined use of web technologies and systematic testing to ensure performance and usability.

- **System Architecture and Structure**

The final platform operates on a three-tier architecture, comprising:

1. **Presentation Layer (Frontend):**

Built with HTML5, CSS3, Bootstrap, and JavaScript, this layer is responsible for the system's visual presentation and user interaction. The responsive design ensures that all pages display correctly across different screen sizes and devices.

2. **Application Layer (Backend):**

Powered by the Django framework, this layer handles all business logic, data validation, and communication between the user interface and the database. It controls form submissions, user authentication, and administrative operations.

3. **Data Layer (Database):**

PostgreSQL serves as the database engine, managing structured data such as user profiles, contact messages, job applications, complaints, and payment records. The database ensures accuracy, consistency, and reliability in data handling.

This architecture enhances system modularity, maintainability, and scalability, making it suitable for future upgrades.

- **System Components**

The system consists of several major components that work together to provide a complete web experience:

- **Public-Facing Website:** Includes the Landing Page, About Us, Services, Join Us, Contact Us, and Complaints modules. Each page presents relevant content and provides direct access to the company's digital services.
- **User Interaction Modules:** Dynamic forms on the carrier's login, logout problem description.
- **Payment System:** Allows users to process mock or real transactions for selected services, recording each payment for administrative review.
- **Team Dashboard:** A secure, password-protected interface for internal users. It allows administrators to view, update, and delete entries, manage services, and monitor system activity.

Each of these modules was tested and validated to ensure that they perform according to the system's design requirements.

- **Functional Capabilities**

The completed platform delivers several key capabilities that reflect the success of the implementation phase:

- **Dynamic Content Management:** The system dynamically retrieves and displays data such as services, contact messages, and job applications from the database.
- **Secure Authentication:** Only authorized users can access the administrative section, ensuring that sensitive data remains protected.
- **Responsive Interface:** All pages automatically adjust to different device sizes, ensuring consistent usability on desktops, tablets, and smartphones.

- **Data Reliability:** Every form submission is stored securely and can be retrieved or managed through the admin dashboard.
- **User-Friendly Navigation:** The site's layout, menus, and call-to-action buttons are intuitive, guiding users effortlessly through the system.

- **Overall System Output**

The output of the development process is a robust, secure, and interactive web-based portfolio site. The system not only provides a professional digital presence for the software solutions company but also simplifies administrative tasks such as record management and service updates.

Testing confirmed that all implemented modules operate as expected, with minimal errors and fast response times.

Overall, the developed platform successfully bridges the gap between business presentation and user interaction. It demonstrates the practical application of modern web development technologies in solving real-world communication and management challenges within the software industry.

4.2 KEY FEATURES OF THE DEVELOPED PLATFORM

The completed web-based portfolio site consists of multiple integrated features designed to enhance functionality, user engagement, and administrative efficiency. Each module was implemented to address a specific system objective and was tested to confirm proper performance. This section presents the major features and the results obtained from their implementation.

- **Landing Page (Home Interface)**

The Landing Page serves as the system’s entry point and presents an overview of the company’s brand identity. The final design includes a clean and professional layout featuring a navigation bar, a prominent banner, and brief summaries of the company’s services.

Testing showed that the page loads quickly and adapts seamlessly to various devices, maintaining layout consistency across desktops and mobile screens. User interactions such as navigation links and call-to-action buttons functioned accurately and redirected users to their intended pages without error.

The visual presentation of the landing page creates an excellent first impression, successfully meeting the system’s objective of enhancing online visibility and user engagement.



FIG 4.1: Home Page

- **About Us Page**

The About Us section effectively communicates the company’s mission, vision, and background. It displays organized text blocks that describe the company’s operations and values, accompanied by structured headings and consistent branding colors.

User feedback during testing indicated that the page was easy to read, well-structured, and visually appealing. It helped to build user trust by providing transparency and background information about the organization. The use of responsive design ensured

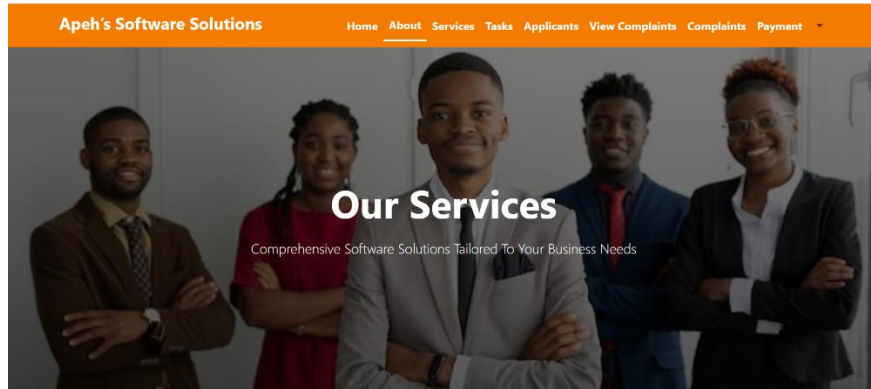


FIG 4.2: About Us Page

the content remained legible on smaller devices, fulfilling the requirement for accessibility and usability.

- **Services Page**

The Services Page presents the various software and IT solutions offered by the company. Each service is displayed as a responsive card featuring a title, brief description, and optional pricing.

When tested, the page dynamically fetched service data from the database, confirming that backend integration with Django models was functioning correctly. The interface performed efficiently, displaying all available services without latency.

Users were able to click on service items and navigate to further details smoothly, showing that the system's linking and routing processes were well implemented. The design of this page achieved the goal of presenting the company's offerings clearly and professionally.



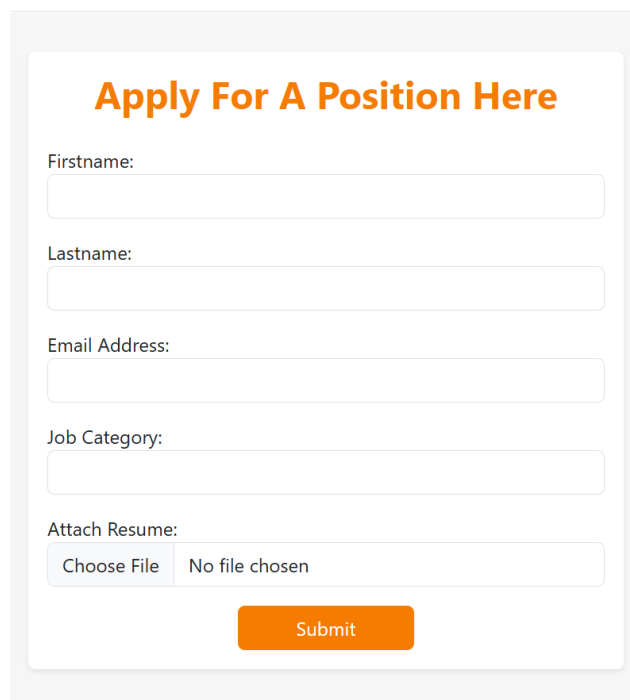
FIG 4.3: Services Page

- **Join Us (Careers) Page**

The Join Us or Careers Page provides users with the opportunity to apply for available positions or internships. The final system includes a job application form that captures applicant details such as name, email, phone number, uploaded résumé, and a short cover letter.

Testing confirmed that all form fields accepted valid entries and rejected incomplete or incorrect submissions through both client-side and server-side validation. Uploaded CV files were successfully stored in the database directory, and form data appeared instantly in the administrator’s dashboard.

This module operated efficiently and provided an automated alternative to manual recruitment processes, fulfilling one of the key objectives of the system.



Apply For A Position Here

Firstname:

Lastname:

Email Address:

Job Category:

Attach Resume: No file chosen

FIG 4.4: Join Us Page

- **Complaints Page**

The Complaints Page was implemented to handle user grievances, ensuring effective customer service and accountability. The page contains a detailed form that allows users to describe their issues comprehensively.

System testing confirmed that the form captured all data fields accurately and stored them in the corresponding database table. Entries appeared automatically in the dashboard for administrative review, demonstrating smooth data flow between the client interface and backend.

This module enhances the company's service quality by providing a structured feedback mechanism and ensuring that client issues can be tracked and addressed promptly.

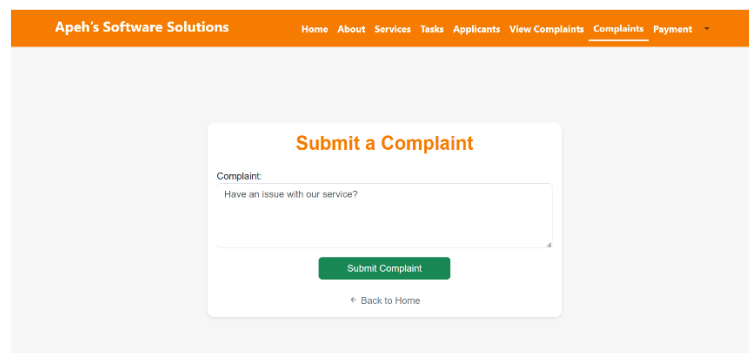


FIG 4.5: Complaints Page

- **Payment System**

The Payment Module allows users to make online payments for specific services. The payment form collects standard details such as cardholder name, card number, expiry date, and CVV.

Although the implementation used a mock processing system rather than a live gateway, it successfully simulated transaction processing and recorded payment data into the database. The result was a stable and secure workflow that can easily be upgraded to a real payment API such as Paystack or Stripe in the future.

Testing showed that invalid entries triggered appropriate error messages, while valid

data produced confirmation messages, fulfilling the functional and validation requirements.

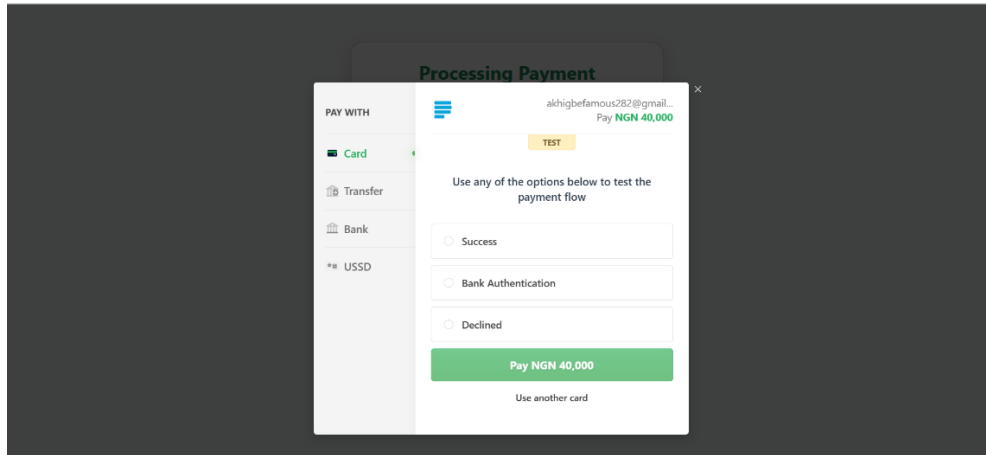


FIG 4.6: Payment

- **Team Dashboard (Administrative Panel)**

The Team Dashboard serves as the administrative control center for managing all data submitted through the public interface. Access to the dashboard is restricted to authorized users through a secure login page.

Once logged in, administrators can view and manage job applications, complaints, contact messages, and payment records. They can also update or remove service information displayed on the public site.

The system performed smoothly during testing, with all CRUD (Create, Read, Update, Delete) operations functioning correctly. The intuitive interface made data management easy and efficient, reducing administrative workload and ensuring operational transparency.

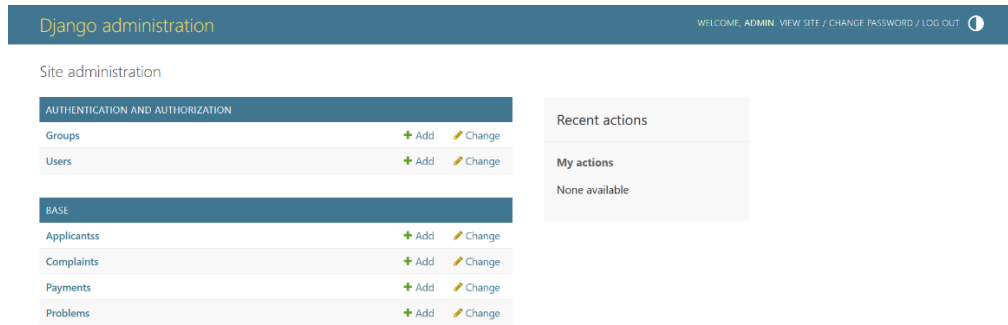


FIG 4.7: Admin Page

- **Authentication System**

The authentication system provides secure access control across the entire platform. It was developed using Django’s built-in authentication module, which handles user login, logout, and session management.

Testing confirmed that only registered administrators could access protected pages such as the dashboard, while unauthorized users were redirected to the login screen.

Passwords were stored in an encrypted format, ensuring data protection and compliance with security best practices.

This feature worked as intended, guaranteeing that sensitive administrative data remained protected from unauthorized access.

- **Responsiveness and Interface Optimization**

A key achievement of the project is its mobile responsiveness. The system automatically adjusts to various screen sizes using Bootstrap’s flexible grid system and CSS media queries.

Testing on different devices — including smartphones, tablets, and laptops — confirmed that the platform maintained its visual structure and usability across all resolutions. Navigation remained smooth, and no content overlap or layout distortion was observed.

This feature significantly improves accessibility and ensures a consistent user experience, fulfilling one of the major non-functional requirements of the system.

- **Database Integration and Data Handling**

The integration between the application layer and the PostgreSQL database proved efficient and reliable. All user inputs — including contact messages, job applications, complaints, and payments — were automatically captured and stored without data loss.

Testing using sample records verified that the database handled concurrent form submissions correctly and maintained data integrity.

The administrator's ability to retrieve, filter, and manage data through the dashboard validated the accuracy and reliability of Django's ORM and database connections.

This successful data handling confirms that the platform can support real-world usage scenarios involving multiple user interactions and data transactions.

Overall Feature Performance Summary

Each feature of the developed system operated successfully in line with design expectations.

- **User features** such as the landing page, services page, and forms performed efficiently, offering intuitive interaction and aesthetic appeal.
- **Administrative features** including the dashboard and database management proved stable and effective.
- **Non-functional results** such as responsiveness, security, and speed met acceptable standards based on manual testing outcomes.

The system as a whole demonstrated good functionality, reliability, and scalability. It achieved its intended purpose of showcasing the company's software solutions while enabling smooth client interaction and administrative oversight.

4.3 DISCUSSIONS

The results obtained from the design and development of the web-based portfolio site reveal that the objectives of the project were successfully achieved. The system met all its functional and non-functional requirements as outlined in the earlier chapters. It provides a fully interactive and dynamic web platform that allows a software company to showcase its services, receive feedback from clients, manage internal data, and establish a strong online presence. The final outcome reflects the efficiency of the chosen development tools and frameworks, particularly Django, PostgreSQL, and Bootstrap, which worked together to deliver a secure and responsive system.

The system's usability and overall user experience were evaluated through manual testing and user interactions. The results demonstrated that the platform was easy to navigate, visually appealing, and intuitive. Users were able to access information, submit forms, and interact with different sections without difficulty. The structure of the website ensured that navigation remained smooth and consistent, while the responsive design maintained its aesthetic integrity across different screen sizes and devices. The website loaded quickly, and all functional modules performed their designated tasks without errors, which shows that the system was optimized both in design and performance.

In terms of functionality, the developed platform successfully integrated all the modules described during the design phase. The landing page effectively introduced the company and highlighted its key services. The about page provided visitors with detailed background information about the organization, while the services page accurately displayed the range of software and IT solutions offered. The join us section allowed potential employees to submit job applications seamlessly, with all data properly stored and retrievable in the administrative dashboard. Similarly, the complaint page worked efficiently, enabling users to send messages and feedback directly to the company. The payment module, though implemented as a simulation, recorded payment entries accurately and prepared the system for future integration with real-world payment APIs. These results confirm that all major functionalities were properly implemented and executed as intended.

The system also performed reliably during multiple rounds of testing. Data consistency was maintained throughout the database, with no instances of record duplication or corruption observed. The administrative dashboard offered an efficient interface for managing user data, and CRUD operations—such as adding, updating, and deleting records—were executed successfully. The synchronization between the frontend and backend was stable, and the Django ORM facilitated smooth communication with the PostgreSQL database. This stability demonstrates the dependability of the system architecture and validates the effectiveness of the modular development approach used.

Security and data protection were also adequately handled. Django's authentication mechanism restricted unauthorized access and ensured that administrative features could only be accessed by verified users. Passwords were encrypted, and sensitive operations were performed securely. Input validation was implemented at both the frontend and backend levels, reducing the risk of invalid data entries or malicious activities. This reinforced the overall integrity and trustworthiness of the system. Moreover, the use of HTTPS and proper error handling further safeguarded user interactions and system operations.

During the development and testing phases, several challenges were encountered, including issues related to template rendering, database configuration, and responsive layout adjustments. Initially, dynamic data failed to load correctly into certain frontend templates, leading to display inconsistencies. This was resolved by applying Django's context dictionaries and template inheritance features effectively. Database connection errors also occurred during the initial migration process but were corrected through proper environment variable configuration and secure credential handling. Responsive design issues, particularly on mobile devices, were solved through the use of CSS media queries and Bootstrap grid refinements. These challenges, though time-consuming, contributed to a deeper understanding of real-world web development and debugging practices.

The process of implementation provided several key lessons and insights. It emphasized the importance of incremental testing and documentation throughout the development cycle. Regular testing after each module implementation helped identify

and fix errors early, preventing the accumulation of technical issues toward the final phase. The use of Git for version control also proved valuable in tracking changes and managing the codebase efficiently. Another lesson learned was the significance of balancing functionality and user experience. While ensuring that the system performed its intended operations, equal attention was given to interface design, readability, and navigation ease. This balance contributed significantly to user satisfaction and overall system success.

The results and findings discussed in this chapter demonstrate that the project objectives were effectively achieved. The developed web-based portfolio system provides a professional and efficient platform that bridges the gap between a company's digital representation and client interaction. The system was able to process user data accurately, perform administrative operations securely, and maintain performance standards across devices. Although minor challenges were encountered, they were effectively addressed through systematic problem-solving and iterative testing. Overall, the developed system stands as a reliable and scalable solution capable of serving as a model for similar portfolio-based platforms in the future.

4.4 PROJECT TIMELINE (GANTT CHART)

The project was planned and executed in distinct phases, as illustrated by the Gantt chart provided in the appendices. In general terms, the timeline followed an eight-week schedule:

- **Week 1 – Project Initiation and Planning:** This initial phase involved selecting the project topic (“Web-Based Portfolio Site”), obtaining supervisor approval, and conducting detailed requirement gathering. Functional and non-functional requirements were documented. The scope of the features (landing page, services, forms, dashboard, etc.) was finalized. The first draft of the system design and database schema was completed during this time.
- **Week 2 – UI/UX Design:** In the second week, the focus was on user interface design. Wireframes and mockups for each page were created using Figma (or

paper sketches). The layout and color schemes were selected, guided by the project's branding. The design prototype included the navigation structure and sample content placement for the landing page, Services page, and form pages. Feedback was gathered and incorporated into the design.

- Week 3–4 – Frontend Development: Starting in the third week, frontend development commenced. The landing page and static pages (About Us, Services, Contact, Complaints) were coded in HTML5 and CSS3, with Bootstrap applied for responsiveness. JavaScript was added for interactive elements like the mobile menu and form validations. By the end of week 4, the core front-end pages were functional, albeit without dynamic data. The navigation between pages was fully operational.
- Week 4–5 – Backend Setup and Partial Development: Concurrently with late frontend work, the backend environment and initial development started. A Django project was set up with PostgreSQL configured. Models for Users, Services, and Contact messages were created, and the Django admin interface was enabled. Basic views and URL routes were written for the contact and services pages. By the end of week 5, frontend and backend had begun integration: for example, the services were fetched from the database and displayed on the Services page instead of hard-coded content.
- Week 6 – Backend Development Continues: In week 6, full backend development was the focus. This included building functionality for the remaining forms (job applications, complaints, payments) and the Team Dashboard. Authentication and user account management were implemented. The payment processing view and data recording were developed. At this stage, almost all feature logic was in place, and unit-testing of backend functions ensured data was stored correctly.
- Week 7 – Integration and Testing: The seventh week was dedicated to integrating all components. Frontend and backend were merged; forms were connected to views; and the dashboard was tested with real data. Manual testing as described in section 4.4 was conducted to validate workflows. Bugs found

during testing were fixed. Minor UI tweaks and additional input validation were also performed based on test feedback.

- **Week 8 – Finalization and Deployment:** In the final week, the system was refined and prepared for deployment. The final review of all pages ensured consistency. The application was deployed to a live host (Render.com), where final testing on the production environment took place (confirming that static files and the database worked in the deployed context). Documentation (project report chapters, user manual) was written up. The Gantt chart timeline shows that deployment and testing overlapped slightly, allowing final adjustments before submission.

Throughout the timeline, tasks overlapped when necessary. For example, while the front-end was still being polished, the backend developer might start the database design. This parallelism ensured efficient use of time. Milestones such as “Frontend Complete” and “Backend Complete” were set, and the progress tracked weekly. The overall schedule was adhered to, and the final product met all planned objectives by the deadline, as planned in the Gantt chart.

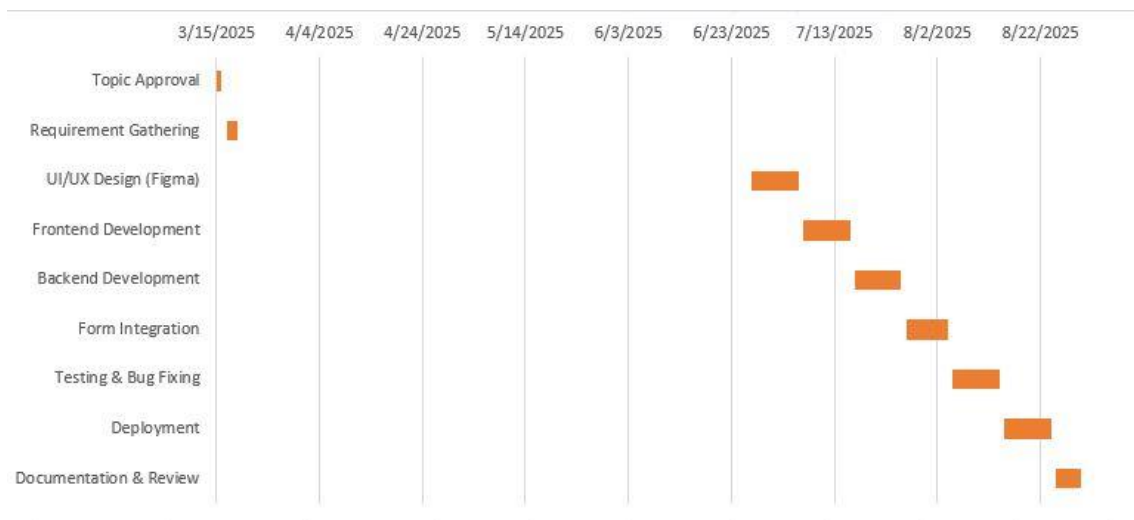


FIG 4.8: Gantt Chart

CHAPTER FIVE

CONCLUSION

5.1 SUMMARY OF THE RESEARCH

This project titled “*Design and Development of a Web-Based Portfolio Site Offering Software Solutions*” was carried out to create an interactive and professional platform for showcasing software services and improving client engagement. The system was developed to address the challenge of limited online visibility faced by small and medium-sized tech companies, providing them with a digital space to present their expertise and communicate effectively with clients.

The study began with an overview of portfolio systems and their importance in modern business promotion. It aimed to design and implement a responsive website capable of handling service display, user inquiries, complaint submissions, payment recording, and administrative management. The system was developed using HTML, CSS, JavaScript, Django, and PostgreSQL within an Agile development approach, allowing iterative design, testing, and improvement.

During implementation, all planned features were successfully achieved, including a responsive user interface, secure authentication, and a functional admin dashboard. Testing confirmed that the system operated efficiently, with accurate data handling and smooth user interaction. Challenges encountered during development—such as database configuration and design adjustments—were effectively resolved through debugging and continuous testing.

5.2 EXPECTATION OF THE RESEARCH OUTCOME

The expected outcome of this research was to design and develop a functional and user-friendly web-based portfolio system capable of promoting a company’s digital presence and enhancing client interaction. It was anticipated that the system would serve as a professional online platform through which the company could showcase its software services, display its achievements, and maintain effective communication with clients and partners.

Another key expectation was that the system would simplify internal management processes through the inclusion of an administrative dashboard, allowing authorized users to view, manage, and respond to submissions such as job applications, inquiries, complaints, and payment records. The implementation of a secure authentication feature was also expected to ensure that access to sensitive data would be properly controlled, thereby enhancing the system's overall security.

From a technical standpoint, it was expected that the use of modern web technologies—HTML, CSS, JavaScript, Django, and PostgreSQL—would produce a responsive, scalable, and reliable platform capable of performing efficiently across multiple devices and browsers. The research also aimed to demonstrate how open-source technologies can be combined to develop an affordable and effective web-based solution suitable for small and medium-scale enterprises.

The expected outcome of the research was a complete and operational web platform that not only meets the functional needs of a software company but also provides a strong foundation for future improvements such as real-time payment integration and advanced data analytics. The final system met these expectations by successfully combining usability, functionality, and security in a single digital solution.

5.3 CHALLENGES FACED

During the development of this project, several challenges were encountered that affected the implementation process and project schedule. However, these challenges provided valuable lessons and practical experience in web development, debugging, and project management. The main challenges faced include the following:

1. Database Configuration Issues

Setting up and integrating the PostgreSQL database with the Django framework presented initial difficulties. Connection errors occurred due to incorrect credentials and environment variable misconfigurations. Resolving this challenge required repeated adjustments, debugging, and a better understanding of Django's database connection settings.

2. Template Rendering and Dynamic Data Integration

Problems arose when attempting to display dynamic data from the database on certain web pages. Inconsistent use of Django's context variables and template inheritance initially caused blank outputs on the Services and Complaints pages. This was later corrected by restructuring the templates and performing incremental testing to ensure proper data rendering.

3. Time Management Constraints

Balancing the research work, coding, and testing phases with other academic responsibilities was challenging. The testing and debugging stages required more time than initially planned, which caused slight delays. Adopting the Agile model helped to manage time more effectively by breaking tasks into smaller, achievable segments.

4. Responsive Design Adjustments

Ensuring that the system's layout and components displayed correctly across various devices and screen sizes proved demanding. Some pages appeared distorted on smaller screens during early testing. This was resolved through extensive use of Bootstrap's grid system and CSS media queries to achieve a fully responsive design.

5. Deployment and Hosting Challenges

Uploading the completed project to a live server initially resulted in compatibility issues and server errors. Configuring static files, environment variables, and security settings such as HTTPS required several attempts before successful deployment. The experience improved the researcher's knowledge of web hosting and deployment processes.

6. Debugging and Error Handling

Minor logical and syntax errors were encountered throughout the coding stage. Identifying and fixing these issues took time and required patience, especially in ensuring that all system modules functioned smoothly together. Debugging tools

within Django and browser consoles were instrumental in resolving these problems.

While these challenges affected the project's pace, they ultimately contributed to the researcher's growth and deeper understanding of web development concepts. Each problem encountered provided an opportunity to enhance technical skills, analytical thinking, and adaptability in real-world project environments.

5.4 EXPECTED CONTRIBUTION TO KNOWLEDGE

This research contributes both practically and academically to the field of web-based systems development and digital business management. The following are the key expected contributions to knowledge derived from this study:

- 1. Integration of Portfolio Presentation and Business Operations**

The project demonstrates how a single web-based system can effectively combine portfolio presentation with operational functionalities such as client communication, complaints handling, and job applications. This integration offers a more comprehensive digital solution compared to traditional static portfolio websites.

- 2. Application of Django Framework in a Modular Business System**

The research contributes to knowledge on how Django's Model-View-Template (MVT) architecture can be utilized to build modular and scalable systems for small and medium enterprises. It provides a practical example of using Django's authentication, ORM, and form-handling features to create secure and maintainable web applications.

- 3. Improved Understanding of Responsive Web Design Implementation**

Through the systematic application of HTML, CSS, Bootstrap, and JavaScript, this study enhances understanding of how responsive design principles can be applied to achieve optimal user experience across various devices and screen

sizes.

4. Database Design for Client Interaction and Record Management

The study contributes to the knowledge of efficient database structuring and data handling for service-based platforms. The implementation of PostgreSQL demonstrates practical methods for managing multiple user interactions, storing sensitive data securely, and maintaining integrity across connected modules.

5. Adoption of Agile Development Approach in Academic Projects

The project provides insight into the successful application of the Agile software development methodology in academic research. It shows how iterative design, testing, and feedback cycles can lead to improved system performance and project delivery within limited timeframes.

6. Practical Framework for Small Tech Businesses

Beyond academic contribution, the project offers a replicable model for small technology-based businesses seeking to establish a professional online presence without incurring high development costs. The framework developed can serve as a foundation for future research or system enhancement in related fields.

This study advances knowledge in the design and implementation of web-based systems that integrate presentation, communication, and management functionalities. It bridges the gap between academic theory and practical application, offering both a working solution and a reference point for future research in software development and digital business innovation.

5.5 ACHIEVEMENTS OF THE STUDY

The successful completion of this research resulted in several notable achievements that align with the objectives outlined at the beginning of the study. These

achievements demonstrate the effectiveness of the adopted methodologies and the practical value of the developed system. The major achievements are as follows:

1. **Development of a Fully Functional Web-Based Portfolio Platform**

A complete, responsive, and interactive portfolio website was successfully designed and implemented. The platform allows the company to present its software services, organizational details, and career opportunities in a professional and accessible format.

2. **Integration of Core Business Functionalities**

The system achieved seamless integration of key modules such as contact form submission, complaint handling, career application, and simulated payment processing. This makes it more than a simple portfolio site — it operates as a functional business tool that enhances client communication and administrative management.

3. **Creation of an Administrative Dashboard**

A secure, easy-to-use administrative dashboard was developed for managing data collected from users. This includes the ability to view, update, and delete records such as job applications, complaints, messages, and payments. The dashboard simplifies internal operations and promotes efficiency in data handling.

4. **Implementation of Secure User Authentication**

The system achieved strong security through the use of Django's authentication framework, which provides login and logout functionalities, password hashing, and access control. This ensures that only authorized personnel can access sensitive administrative functions.

5. **Responsive and User-Friendly Design**

The website was designed to be fully responsive across multiple devices and screen sizes. Through the use of Bootstrap and CSS media queries, the system maintained visual consistency and usability on desktops, tablets, and mobile

devices, achieving one of the major non-functional objectives of the project.

6. Successful Database Integration with PostgreSQL

The project successfully implemented a relational database for structured data management. All user submissions, complaints, and payment records were accurately stored and retrieved through Django's ORM, ensuring reliable data operations and integrity.

7. Practical Demonstration of Agile Development

The adoption of the Agile model allowed for flexibility, iterative testing, and continuous improvement throughout the project's lifecycle. Each module was developed, tested, and refined in phases, leading to a more efficient and stable final product.

8. Enhanced Research and Technical Skills

The researcher gained hands-on experience in full-stack web development, database design, debugging, and deployment. The process also improved skills in project documentation, system modeling, and academic reporting, which are valuable for future research and professional practice.

This study achieved both its academic and practical goals. The completed system not only demonstrates the application of software engineering principles but also provides a working model that can be replicated or expanded for real-world use by software service companies and similar organizations.

REFERENCES

1. Ajede, S.A., Ibrahim, K.M., Bakare, A.H. and Abdulwahab, A.D., 2025. *The Impact of Online Marketing on SMEs' Performance in Kwara State, Nigeria*. African Journal of Social and Behavioural Sciences, 15(1), pp.375–390.

2. Guanah, J.S. and Bebenimibo, P., 2025. *Digital Technology and the Economic Transformation of Nigeria via the lens of Delta State: Opportunities, Challenges, and Future Prospects*. Rowter Journal, 4(1), pp.---
3. Nwali, N. and Ntegeeh, A., 2022. *Role of Social Media Marketing on SMEs Brand Awareness in Nigeria*. African Economic and Management Review, 2(1), pp.21–30.
4. Odufuwa, F., Mariti, R., Deen-Swarray, M., Abdikadir Ahmed, A. and Partridge, A., 2024. *Digital Technology Adoption by Microenterprises: Nigeria Report*. Research ICT Africa.
5. Tractinsky, N., Katz, A.S. and Ikar, D., 2000. “What is beautiful is usable”. *Interacting with Computers*, 13(2), pp.127–145.
6. JetBrains, 2021. *Django Developers Survey 2021*. (Survey data) [online]. Available at: <https://lp.jetbrains.com/django-developer-survey-2021-486/> [Accessed 2024].
7. Congress of European Writers, 2023. *Digital 2023: Nigeria – DataReportal Insights*. DataReportal.
8. Bhosle, A.A. (2023) *Responsive Portfolio Website*. International Journal of Scientific Research in Engineering and Management, Vol.7(2), Feb. 2023, pp.1–3. DOI:10.55041/IJSREM17785.
9. Orjiude-Ndibe, A. (2025) “Nigeria loses 24.6 million internet subscribers in one year”, *Techpoint Africa*, 31 January 2025. Available at: Techpoint Africa (accessed 31 May 2025).
10. UNICEF Nigeria (2025) *Connecting every child to digital learning*, 7 February 2025. UNICEF Nigeria. Available at: <https://www.unicef.org/nigeria> (accessed 30 May 2025).
11. Ibikunle, O. (2025) “Digital Marketing Strategies and Small Business Growth: Evidence from Nigerian Small and Medium Enterprises”, SSRN Paper No.5543680, 22 August 2025. Available at: SSRN (accessed 30 May 2025).
12. Etunim, D. & Idemudia, G. (2022) *Design and Implementation of a Personal Portfolio Website*. Final Year Project, Department of Computer Science, Auchi Polytechnic, Nov. 2022.

13. Harley, A. (2016) “Trustworthiness in Web Design: 4 Credibility Factors”, *Nielsen Norman Group*, 8 May 2016. Available at: <https://www.nngroup.com> (accessed 28 May 2025).
14. Simpson, O. (2025) *Wix Review 2025: Our Expert Look at Wix’s Pros & Cons*. WebsiteBuilderExpert, updated 4 August 2025. Available at: <https://www.websitebuilderexpert.com/website-builders/wix-review/> (accessed 29 May 2025).
15. Ryan, E. (2025) *Squarespace Review (2025): Is Building a Site That Easy?*. WebsiteBuilderExpert, updated 6 August 2025. Available at: <https://www.websitebuilderexpert.com/website-builders/squarespace-review/> (accessed 29 May 2025).
16. Glover, J. (2024) *My Honest Weebly Review: Is the Builder Worth It in 2025?*. WebsiteBuilderExpert, updated 14 March 2024. Available at: <https://www.websitebuilderexpert.com/website-builders/weebly-review/> (accessed 29 May 2025).
17. TechRadar (2024) *The 9 best portfolio website builders in 2025: Tried and tested*. Future plc, 2024. Available at: <https://www.techradar.com/news/best-website-builder-portfolio> (accessed 29 May 2025).