



Tribhuvan University

Institute of Science and Technology

Project Report on

Frontend Intern at Global Square I.T. Company Pvt. Ltd.

Submitted to:

Department of Bachelor of Science in Computer Science and Information
Technology (BSc.CSIT)
National College of Computer Studies

*In partial fulfillment of the requirement for the degree of
Bachelor of Science in Computer Science and Information Technology
(BSc.CSIT)*

Submitted by:

Amardeep Limbu [28860/078]
Regd. No.: 5-2-551-64-2021

Under the supervision of
Mr. Radha Krishna Gajurel

January, 2026

SUPERVISOR'S RECOMMENDATION

This is to certify that the internship report entitled "**Frontend Internship at Global Square IT Company Pvt. Ltd.**" has been prepared by **Amardeep Limbu [28860/078]** under my supervision and guidance.

The report reflects the student's practical experience, technical skill development, and the successful application of computer science principles in a professional environment. I have monitored the progress of this internship and the subsequent documentation throughout the semester.

I hereby recommend that this report, prepared in partial fulfillment of the requirements for the degree of **Bachelor of Science in Computer Science and Information Technology (BSc.CSIT)**, be accepted and processed for the final evaluation.

Mr. Radha Krishna Gajurel
Project Supervisor
National College of Computer Studies
Paknajol, Kathmandu

CERTIFICATE OF APPROVAL

This is to certify that the report is evaluated and recommended to the Department of Computer Science and Information Technology for acceptance of the internship report entitled "**Frontend Engineering Internship at Global Square I.T. Company Pvt. Ltd.**" submitted by **Amardeep Limbu [28860/078]** in partial fulfillment for the degree of Bachelor of Science in Computer Science and Information Technology (B.Sc.CSIT), Institute of Science and Technology (IOST), Tribhuvan University.

Signature of Supervisor

Mr. Radha Krishna Gajurel

Faculty Member

Computer Science and Information
Technology

National College of Computer Studies
Paknajol, Kathmandu

Signature of Coordinator

Ramesh Phuyal

Coordinator of CSIT

Department of Computer Science and
Information Technology

National College of Computer Studies
Paknajol, Kathmandu

SIGNATURE of Internal Examiner

SIGNATURE of External Examiner

ACKNOWLEDGMENT

I wish to express my sincere appreciation to **Global Square I.T. Company Pvt. Ltd.** for the opportunity to undertake my professional internship within their esteemed organization. The exposure to a collaborative and technologically advanced work environment has been pivotal in bridging the gap between my academic knowledge and the professional landscape.

I am deeply grateful to the **Department of Computer Science and Information Technology** at **National College of Computer Studies** for their consistent support. I owe a special debt of gratitude to our Program Coordinator, **Mr. Rajan Paudel**, whose efficient coordination and constant encouragement were essential for the successful completion of this internship.

I would like to extend my heartfelt thanks to my project supervisor, **Mr. Radha Krishna Gajurel**, for his invaluable guidance. His academic insights and constructive critiques throughout the semester were instrumental in the refinement of this report.

I am particularly thankful to my mentors at the organization, **Mr. Nabin Shrestha** and **Mr. Sushant Luitel**. Their expert technical mentorship and willingness to share their expertise allowed me to navigate complex challenges and contribute effectively to the company's projects.

Finally, I am thankful to my family, friends, and colleagues for their motivation and support during my academic tenure. Their shared insights and encouragement made this entire learning experience truly enriching.

Sincerely,

Amardeep Limbu

Roll No: 28860/078

TU Regd. No: 5-2-551-64-2021

ABSTRACT

This report outlines the design and development of responsive web applications implemented during a frontend engineering internship. The primary objective was to construct scalable, user-centric interfaces through modern component-based architectures using React.js and Next.js. The development scope encompassed diverse applications, including e-commerce platforms and online examination systems, necessitating robust state management and cross-device compatibility. Key technical implementations involved strict type safety via TypeScript, schema validation, and seamless RESTful API integration to ensure data integrity and system reliability. Furthermore, the project emphasizes professional software engineering practices, including version control, modular code organization, and design-to-code translation. The outcomes demonstrate significant enhancements in application performance, usability, and maintainability, validating the effectiveness of integrating advanced frontend methodologies to deliver industrial-grade web solutions.

KEYWORDS: *Frontend Development, Web Applications, React.js, Component Architecture, API Integration*

TABLE OF CONTENTS

SUPERVISOR'S RECOMMENDATION	i
CERTIFICATE OF APPROVAL	ii
ACKNOWLEDGMENT	iii
ABSTRACT	iv
LIST OF FIGURES	vii
LIST OF TABLES	viii
LIST OF ABBREVIATIONS	ix
CHAPTER 1: INTRODUCTION	1
1.1 Introduction	1
1.2 Problem Statement	1
1.3 Objective	1
1.4 Scope and Limitations	2
1.4.1 Scope	2
1.4.2 Limitations	2
1.5 Report Organization	2
CHAPTER 2: ORGANIZATION DETAILS AND LITERATURE REVIEW	4
2.1 Introduction to Organization	4
2.2 Organizational Hierarchy	5
2.3 Working Domains Of Organization	5
2.3.1 Key Working Domains of the Organization	5
2.4 Description of Intern Department	6
2.5 Literature Review	6
CHAPTER 3: INTERNSHIP ACTIVITIES	9
3.1 Roles and Responsibilities	9
3.1.1 Roles	9
3.1.2 Responsibilities	9
3.2 Weekly Log	9

3.3 Tasks/Activities Performed	11
CHAPTER 4: CONCLUSION AND LEARNING OUTCOMES	39
4.1 Conclusion	39
4.2 Learning Outcomes	39
REFERENCES	41
APPENDIX	42

LIST OF FIGURES

1	Organization Logo of Global Square I.T Company	4
2	Organization Hierarchy	5
3	Code Snippet of Hoslog Affiliate Landing Page	12
4	Hoslog Affiliate Landing Page Hero Section	13
5	Hoslog Affiliate Landing Page Features Section	13
6	Hoslog Affiliate Landing Page FAQ Section	14
7	Hoslog Affiliate Landing Page Footer Section	14
8	Code Snippet of Custom Hook for Scroll to Top	16
9	Code Snippet of Blog Site Clone	17
10	Blog Site Homepage	18
11	Blog Site Signup Page	18
12	Code Snippet of Online Exam Site Routing	20
13	Online Exam Site Listening Section	21
14	Online Exam Site Speaking Section	22
15	Online Exam Site Writing Section	23
16	Code Snippet of Online Exam Site Summary Section	24
17	Online Exam Site Reading Section	25
18	Code Snippet of Square Japan inquiry Page	27
19	Code snippet of the Square Japan inquiry page (full view)	28
20	Code Snippet of Square Japan Landing Page	29
21	Tapaiko Bazar Admin Login Interface	31
22	Tapaiko Bazar Vendor Form Module	32
23	Vendor Table Overview	33
24	Product Details and Management View	33
25	Product Management Dashboard	34
26	Filtered Product Data View	34
27	Client Portal Product Cart Interface	35
28	Website Review and Rating Component	35
29	Tapaiko Bazar Product Management UI	36
30	Product Slider with Media Gallery	37
31	Variant Handling in Product Creation	38

LIST OF TABLES

1	Contact Details of Organization	5
2	Internship Details	6
3	Weekly Tasks Performed During Internship	9

LIST OF ABBREVIATIONS

API	Application Programming Interface
CSS	Cascading Style Sheets
HTML	HyperText Markup Language
JS	JavaScript
JSON	JavaScript Object Notation
UI	User Interface
UX	User Experience
OTP	One-Time Password
SPA	Single-Page Application
REST	Representational State Transfer

CHAPTER 1: INTRODUCTION

1.1 Introduction

In the beginning, one of the projects that took part was developing the landing page for Affilater Hoslog at Global Square IT Company. So, my major objective was going to convert the predefined Figma design into a visually fully functional website. The landing page was developed using React.js and vanilla CSS, paying close attention to responsive layouts and mobile compatibility through CSS media queries. This helped me have real exposure to translating design specifications into a dynamic and interactive user interface.

Another important project was the creation of a landing page for Square Japan. More precisely, it included interactive features that served as a recruitment page and inquiry form. The Next.js library was applied to build a landing page with Tailwind CSS styling that kept the views consistent, maintainable, and responsive.

Contributed to the full-featured, collaborative e-commerce application. This included responsibilities such as designing and implementing the front-end UI for both customer and admin portals. Admin side comprises a dashboard developed along with a sidebar and data visualization charts using Recharts for actionable insights; on the client side, key pages such as login page, product listing pages were developed.

1.2 Problem Statement

With the increasing demand for seamless online experiences, organizations need web applications that are both visually attractive and functionally robust. Frontend development involves several difficulties, such as converting design mockups into functional interfaces, managing dynamic data, ensuring cross-device compatibility, and maintaining code quality for long-term scalability. The internship mainly focused on addressing these challenges by working on real-world projects, implementing features like authentication, dynamic product displays, responsive layouts, and interactive components, thereby bridging the gap between academic learning and practical frontend development experience.

1.3 Objective

The goals of the internship were as follows:

- To follow the industry's best practices in front-end development by writing clean, modular, and maintainable code, performing UI testing, and using version control tools to support collaboration

- To integrate frontend components with backend services, implement dynamic features, and enhance application performance, usability, and maintainability through real-world project development.

1.4 Scope and Limitations

1.4.1 Scope

The scope of the E-commerce web application developed during the internship includes the following:

- The internship involved creating front-end interfaces for various web applications, including landing pages and e-commerce platforms, utilizing modern technologies such as React.js, Next.js, CSS, and Tailwind CSS.
- The work emphasized responsive design, user experience, integration with backend APIs, and development of dynamic features to improve functionality and interactivity.

1.4.2 Limitations

Despite its functionality and practical implementation, the E-commerce web application developed during the internship faces the following technical and operational limitations:

- The internship was mainly centered on frontend development, providing limited experience with advanced backend processes, database management, and server-side optimization.
- Certain projects were restricted by predefined designs, tight deadlines, and dependency on backend-developed APIs, which constrained full end-to-end control over the system.

1.5 Report Organization

The report is organized in the systematic pattern where each chapter includes the subtopics to support the main title of the report. The report is organized as follows:

- Chapter 1: Introduction**

The chapter initiates with the introduction of the whole report, systematically partitioned into four distinct subchapters. It includes introduction of the internship project, problem statement, objectives, scope and limitation.

- Chapter 2: Organizational Review and Literature Review**

This chapter offers a concise overview of organization involved in the project and includes a review of relevant literature related to the project's domain.

- **Chapter 3: Internship Activities**

This chapter includes the roles, responsibilities and technical activities undertaken during the internship in detail. It also covers descriptions of the projects involved and a comprehensive list of tasks and activities completed.

- **Chapter 4: Conclusion and Learning Outcomes**

This chapter summarizes the project's architectural design, details the data utilized, and provides insights into the overall execution process of the project.

CHAPTER 2:

ORGANIZATION DETAILS AND LITERATURE REVIEW

2.1 Introduction to Organization

Global Square IT Company is a technology-focused software development firm located at Nil Kamal Marg, New Baneshwor, focused on delivering customized software solutions to support business transformation. The company specializes in designing and developing scalable and efficient software applications for a wide range of industries, including finance, healthcare, and retail. By leveraging modern development practices and a skilled technical workforce, Global Square aims to address complex business challenges through reliable and high-performance software systems.

A key characteristic of Global Square IT Company is its emphasis on building bespoke and client-oriented solutions rather than relying on generic software products. The organization adopts modern technologies such as C#, .NET, ReactJS, and cloud-based services to develop applications that are flexible, secure, and adaptable to evolving business requirements. The development process follows structured workflows that prioritize code quality, performance optimization, and seamless user experience, ensuring that the delivered solutions align closely with client expectations.

Within the broader context of the rapidly evolving technology sector, Global Square IT Company positions itself as a growth-driven organization focused on continuous improvement and innovation. By integrating emerging technologies and maintaining a strong collaborative development culture, the company supports long-term digital transformation for its clients. This approach enables Global Square to deliver sustainable value, enhance productivity, and maintain a competitive edge in an increasingly digital and competitive marketplace.



Figure 1: Organization Logo of Global Square I.T Company

Table 1: Contact Details of Organization

Address	Nil Kamal Marg, New Baneshwor, Nepal
Contact Number	976-3596355
Email	info@globalsquareit.com
Website	https://www.globalsquareit.com

2.2 Organizational Hierarchy

The following diagram shows the flat hierarchical structure that Global Square IT Company uses, the key role consists of:

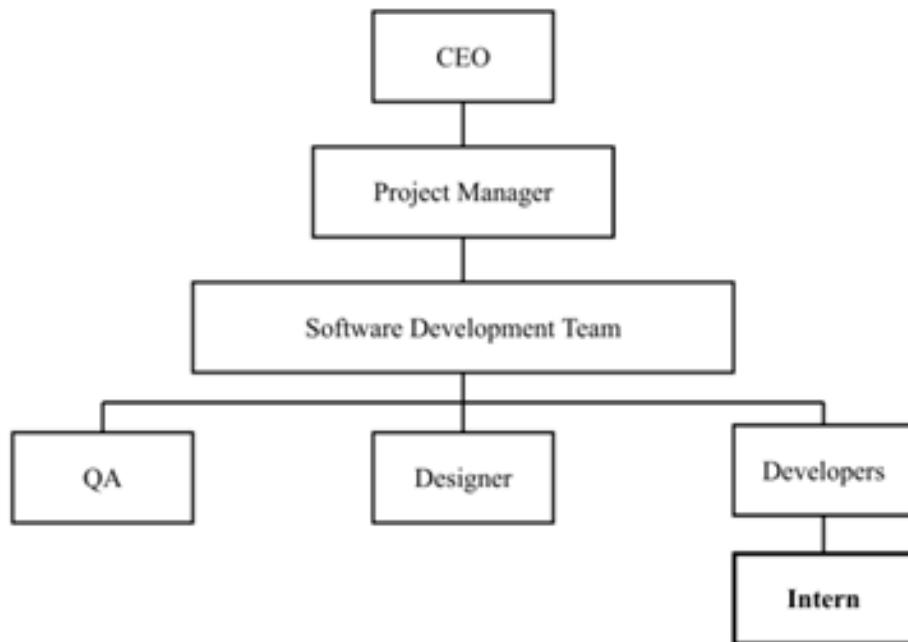


Figure 2: Organization Hierarchy

2.3 Working Domains Of Organization

2.3.1 Key Working Domains of the Organization

The key working domains of the organization include:

1. Web and Application Development

Global Square IT Company specializes in designing and developing modern, scalable web and software applications. This domain includes:

- **Frontend Development:** Building responsive and interactive user interfaces using technologies such as React.js, Next.js, and Tailwind CSS.
- **Backend Development:** Developing secure and efficient server-side applica-

tions using C#, .NET, and RESTful APIs.

- **Full-Stack Solutions:** Integrating frontend and backend components to deliver complete, high-performance applications tailored to client requirements.

2. UI/UX Design and Digital Solutions

The company places strong emphasis on user-centric design and digital transformation. This domain includes:

- **UI/UX Design:** Creating visually appealing, intuitive, and user-friendly interfaces to enhance user engagement and accessibility.
- **System Integration:** Integrating third-party services, APIs, and cloud-based solutions to improve functionality and scalability.
- **Maintenance and Optimization:** Providing continuous support, performance optimization, and updates to ensure long-term reliability and efficiency of software systems.

2.4 Description of Intern Department

At Global Square IT Company Pvt. Ltd., the Intern Department plays a vital role in connecting academic knowledge with advanced industry practices. As the company works with international clients, particularly from Japan, the department operates in a technically demanding, research-focused, and dynamic environment that prepares interns for real-world challenges.

Table 2: Internship Details

Role	Frontend Intern
Department	Frontend Department
Mentor	Nabin Shrestha, Sushant Luitel
Project	Hoslog Affiliater Landing Page, Ecommerce Website, Square Japan Landing Page
Duration	3 Months
Start Date	22nd September 2025
End Date	26th December 2025
Working Hours Per Day	10 AM - 6 PM (Monday to Friday)

2.5 Literature Review

Modern web applications heavily rely on the fundamental principles of JavaScript, particularly asynchronous programming, execution contexts, and event-driven architectures. Resources like W3Schools and JavaScript.info provide comprehensive explanations of these concepts, covering topics such as promises, async/await patterns, closures, and the call

stack. Understanding these fundamentals is critical for handling complex application behavior, managing concurrent tasks, and ensuring that code execution remains predictable and efficient. Mastery of these concepts forms the foundation for integrating advanced frameworks and libraries while adhering to modern ECMAScript standards, allowing developers to build robust, high-performance applications. JavaScript.info (2024) W3Schools (2024)

Managing data effectively is a key challenge in modern web development. TanStack Query addresses this by providing a server-state management solution that eliminates much of the manual client-side state tracking. Unlike traditional state management approaches, which often require repetitive boilerplate code for fetching, caching, and updating data, TanStack Query automates these processes. It offers built-in caching strategies, background data synchronization, and query invalidation, ensuring that the user interface always reflects the most up-to-date server state. By shifting the focus from local state management to server-driven state, applications can achieve improved performance, consistency, and maintainability. TanStack (2024a)

Effective routing is essential for building dynamic, single-page applications. TanStack Router enhances traditional routing by treating the URL as the primary source of truth, enabling type-safe state management and predictable navigation flows. This approach allows for consistent and bookmarkable application states, while ensuring that each page or component has access to the required data and parameters. TanStack Router integrates closely with server-state management tools, enabling developers to synchronize route transitions with data fetching and validation. As a result, applications built with this architecture maintain both functional consistency and a smooth user experience across complex navigation scenarios. TanStack (2024b)

Ensuring data correctness is a critical aspect of web application reliability. Zod is a TypeScript-first schema validation library that provides runtime validation of data structures. It allows developers to define strict schemas for API responses, form inputs, and application state, automatically catching errors before they propagate through the system. Zod's seamless integration with TypeScript also enables strong type inference, reducing the likelihood of runtime errors and improving code safety. By validating data at the boundaries of the system, applications can maintain integrity, prevent unexpected crashes, and provide more reliable user experiences. Zod (2024)

Presenting complex datasets in a clear and efficient manner is a common requirement for web applications. TanStack Table implements a “headless” architecture, separating data-handling logic from visual presentation. This approach allows developers to implement features such as sorting, filtering, grouping, and pagination without imposing any constraints on the UI design. By decoupling logic from interface, TanStack Table enables

high-performance data handling while maintaining complete control over the look and feel of the application. This separation promotes modularity, simplifies testing, and improves maintainability for data-intensive applications.TanStack (2024c) Modern UI development often requires reusable, consistent, and customizable components. Shadcn provides a modular component library designed to integrate seamlessly with modern frameworks while allowing full control over styling and behavior. Its component-driven architecture enables developers to build scalable interfaces, maintain design consistency, and quickly adapt to evolving project requirements. By combining modular UI components with headless libraries for logic, applications achieve both high usability and maintainability, while reducing development time and improving the overall quality of the user interface.shadcn/ui (2024)

CHAPTER 3: INTERNSHIP ACTIVITIES

This chapter details the tasks, responsibilities and technical activities performed during the internship. It includes an overview of assigned duties, a weekly log of activities, and description of projects worked on.

3.1 Roles and Responsibilities

3.1.1 Roles

- Frontend Intern

3.1.2 Responsibilities

During the internship, the following responsibilities were undertaken:

- Designing and developing responsive user interfaces using modern frontend technologies such as React.js and Next.js.
- Converting UI/UX designs into functional web pages while ensuring visual accuracy and cross-browser compatibility.
- Implementing reusable components, managing application state, and organizing frontend code for scalability and maintainability.
- Integrating frontend applications with backend APIs to enable dynamic data handling and real-time functionality.
- Testing, debugging, and optimizing frontend features to improve performance, responsiveness, and overall user experience.

3.2 Weekly Log

Table 3: Weekly Tasks Performed During Internship

Week	Description of Tasks Performed
Week 1 (Sep 22-27)	<ul style="list-style-type: none">• Learned React Basic Fundamentals• Learned best practices in JS programming• Learned tailwind CSS

Week	Description of Tasks Performed
Week 2 (Oct 6-10)	<ul style="list-style-type: none"> • Learned React Hooks: useState and useEffect • Learned Custom Hooks in React and its useCases • Implemented useState and useEffect for state management • Created Custom Hook counter to keep track of count of button pressed
Week 3 (Oct 13-17)	<ul style="list-style-type: none"> • Completed frontend training exercises in React.js, JavaScript, and modern tools • Developed basic components: loading bar, toast notifications, loading skeletons • Learned Git and GitHub for version control • Built a weather application using public API with TanStack Query
Week 4 (Oct 27-31)	<ul style="list-style-type: none"> • Implemented Hoslog Affiliater Landing Page from Figma design • Developed page using React.js and vanilla CSS • Ensured mobile responsiveness and cross-device compatibility using CSS media queries
Week 5 (Nov 3-7)	<ul style="list-style-type: none"> • Identified and resolved UI/UX issues reported by the design team • Fixed layout, responsiveness, and interaction-related bugs • Re-tested website across different devices to ensure stability
Week 6 (Nov 10-14)	<ul style="list-style-type: none"> • Learned React Hook Form and Zod • Implemented form validation using React Hook Form + Zod • Designed and developed a clone of a Blog site • Collaborated with fellow interns using Git for team workflows • Implemented UI features: pagination, scroll to top, image animations
Week 7 (Nov 17-21)	<ul style="list-style-type: none"> • Created a landing page for an online Exam site • Create exam page like Multiple choice question and essay section • Implemented pagination and routing for the webpage

Week	Description of Tasks Performed
Week 8 (Nov 24-28)	<ul style="list-style-type: none"> • Developed Square Japan Landing Page from Figma design • Created Recruitment section • Implemented inquiry forms using React Hook Form + Zod • Integrated Shadcn UI components for cards
Week 9 (Dec 1-5)	<ul style="list-style-type: none"> • Developed Tapaiko Bazar e-commerce website • Fix issue in the Square Japan Landing Page • Built admin portal using React.js + Shadcn UI; client portal using Next.js + Shadcn UI • Created reusable and scalable components • Designed admin vendor table, profile page, single product page, and login page using dummy data
Week 10 (Dec 8-12)	<ul style="list-style-type: none"> • Consumed and integrated backend APIs for dynamic functionality • Implemented admin-side features: login, add-product, profile, vendors • Implemented admin-side APIs for profile, product, and vendor management

3.3 Tasks/Activities Performed

1. Hoslog Affiliate Landing Page

The Hoslog Affiliate Landing Page was developed using React.js in combination with vanilla CSS. The primary objective of the project was to translate a predefined Figma design into a fully functional and responsive web application. Particular emphasis was placed on maintaining visual fidelity to the original design, ensuring cross-device compatibility through the use of CSS media queries, and enhancing overall user experience. The development process incorporated responsive layout techniques and interactive UI components to improve usability. Additionally, the landing page was tested across multiple web browsers to ensure consistent performance and functionality in different browsing environments.

```
1  export const Header = () => {
2    const slider = [
3      smapp,
4      clubair,
5      topdandy,
6      clubacqua,
7      roeo,
8      sinceyou,
9      clubharem,
10    ];
11
12    return (
13      <div className={styles.header}>
14        <div className={styles.navbar}>
15          <img src={logomobile} className={styles["navbar-logo"]} />
16          <button className={styles.button}>
17            アフィリエイトプログラムに参加する(無料)
18          </button>
19        </div>
20        <div className={styles.main}>
21          <div className={styles.content}>
22            <div className={styles.logo}>
23              <img src={logo} className={styles["logo-image"]} />
24            </div>
25
26            <div className={styles["text-content"]}>
27              <div className={styles["text-container"]}>
28                <div className={styles.text}>
29                  ホストクラブの初回ネット予約
30                  <br />
31                  「ホスログ」のアフィリエイトプログラム
32                </div>
33              </div>
34
35              <div className={styles["text-2"]}>
36                ホスロ<span className={styles.highlight}>グ</span>
37                <span className={styles.highlight1}>で</span>
38                <span className={styles["text-green"]}>継続的</span>
39                <span className={styles.highlight2}>な</span>
40                <br />
41                紹介報酬が受け取れる
42              </div>
43            </div>

```

Figure 3: Code Snippet of Hoslog Affiliate Landing Page



Figure 4: Hoslog Affiliate Landing Page Hero Section



Figure 5: Hoslog Affiliate Landing Page Features Section

よくあるご質問

Q. ホスログのアフィリエイトプログラムに参加するにはどうすればいいですか？ ▾

Q. アフィリエイトプログラムに参加するための費用はかかりますか？ ▾

Q. アフィリエイトリンクをどのように取得できますか？ ▾

Q. 誰でもアフィリエイトプログラムに参加できますか？ ▾

Q. どのようにして報酬を得ることができますか ▾

Q. 報酬はどのように支払われますか？ ▾

Q. 自分の紹介したお客様が再訪した場合、報酬はどうなりますか？ ▾

Q. 報酬が正しく支払われていない場合、どうすれば良いですか？ ▾

Q. アフィリエイターの報酬を確認する方法はありますか？？ ▾

Q. 報酬の最低支払額はありますか？ ▾

Q. アフィリエイター紹介プログラムとは何ですか？ ▾

Q. アフィリエイターを紹介する方法は？ ▾

Q. 紹介したアフィリエイターがどれくらいの報酬を稼いでいるか確認できますか？ ▾

Q. 紹介プログラムの報酬はいつ支払われますか？ ▾

Q. アフィリエイトリンクをどのように活用すれば良いですか？ ▾

Q. 特定のホストクラブへの紹介を優先することはできますか？ ▾

Q. アフィリエイトリンクをシェアする際の注意点はありますか？ ▾

Figure 6: Hoslog Affiliate Landing Page FAQ Section



The footer section features the Hoslog logo at the top, followed by a large, bold headline: "\ 登録無料で簡単に広告活動が始められる！ /". Below the headline is a subtext: "ホスログのアフィリエイトプログラムに今すぐご参加ください！". A prominent purple button with white text reads "無料登録をして今すぐ広告活動を始める". At the bottom of the page, there is a green footer bar containing the copyright notice: "© 2019 - 2026 ホスログ".

Figure 7: Hoslog Affiliate Landing Page Footer Section

2. Blog Site Clone

The Blog Site Landing Page comprises three primary features developed using React with TypeScript. The Homepage implements pagination functionality with six articles per page, utilizing React Router's searchParams API to maintain state and Tailwind CSS media queries for responsive design across devices. The ScrollToTop utility component monitors route changes via the useLocation hook and automatically scrolls the viewport to the page top with smooth behavior, enhancing navigation experience in the single-page application. The Signup page integrates schema-based form validation using Zod and React Hook Form, capturing user credentials with real-time error feedback and asynchronous submission handling, providing a seamless user registration experience with confirmation messaging and navigation capabilities.



```
1 import { useEffect } from "react";
2 import { useLocation } from "react-router-dom";
3
4 export const ScrollToTop = () => {
5   const { pathname, search } = useLocation();
6   useEffect(() => {
7     window.scrollTo({ top: 0, behavior: "smooth" });
8   }, [pathname, search]);
9
10  return null;
11 }
```

Figure 8: Code Snippet of Custom Hook for Scroll to Top

```
 1  export const Home = () => {
 2    const [searchParams, setSearchParams] = useSearchParams();
 3    const page_size = 6;
 4    const page = Number(searchParams.get("page")) || 1;
 5    const totalPages = Math.ceil(HomeContent.length / page_size);
 6    const start = (page - 1) * page_size;
 7    const visiblePosts = HomeContent.slice(start, start + page_size);
 8
 9    const goToPage = (pageNumber: number) => {
10      setSearchParams({ page: pageNumber.toString() });
11      window.scrollTo({ top: 0, behavior: "smooth" });
12    };
13
14    return (
15      <div className="w-full flex flex-col justify-center md:flex-row md:gap-8 mx-auto mt-10 px-4 md:px-0">
16        <div className="w-full max-w-7xl lg:w-2/3">
17          <div id="blogarticle">
18            {visiblePosts.map((post, index) =>
19              <PostCard key={index} post={post} />
20            ))}
21          </div>
22
23          <div className="flex items-center justify-center gap-5 my-6">
24            {page > 1 && (
25              <button
26                onClick={() => {
27                  goToPage(page - 1);
28                }}
29                className="px-3 py-2 rounded border bg-red-500 text-white w-20"
30              >
31                Prev
32              </button>
33            )}
34
35            <span className="text-sm">
36              Page {page} of {totalPages}
37            </span>
38
39            {page < totalPages && (
40              <button
41                onClick={() => {
42                  goToPage(page + 1);
43                }}
44                className="px-3 py-2 rounded bg-red-500 text-white border w-20"
45              >
46                Next
47              </button>
48            )}
49          </div>
50        </div>
51        <Sidebar />
52      </div>
53    );
54  };
55};
```

Figure 9: Code Snippet of Blog Site Clone

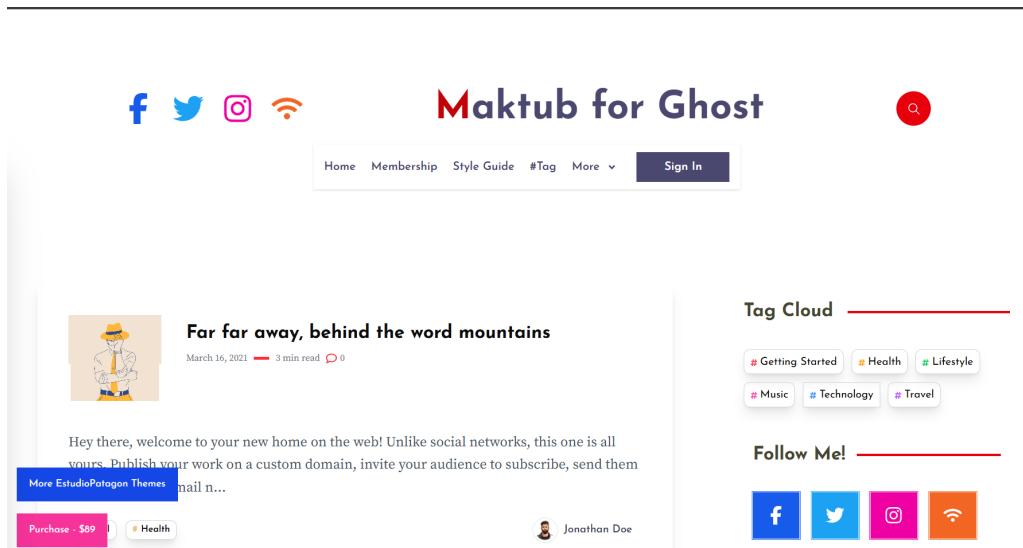


Figure 10: Blog Site Homepage

The screenshot shows the 'Subscribe to Maktub' sign-up page. The page has a clean, modern design with a white background and light gray borders around input fields. It features five input fields: 'Name' (placeholder: 'Enter your name'), 'Email' (placeholder: 'Enter your email'), 'Phone No' (placeholder: 'Enter phone number'), 'Password' (placeholder: 'Create password'), and 'Confirm Password' (placeholder: 'Confirm password'). Each field is accompanied by a label above it.

Figure 11: Blog Site Signup Page

3. Online Exam Site

The Online Exam Application utilizes a structured navigation system built with React Router v6 and TypeScript to ensure a seamless and predictable user journey. At the core of the architecture is a centralized routing component that wraps the entire application in a dedicated exam provider. This setup uses the React Context API to maintain a consistent global state, allowing critical data—such as user responses, remaining time, and overall progress—to persist even as the student moves between different sections of the test.

The navigation logic is designed to guide users through a linear examination flow, starting with an automatic redirect from the root path to the first assessment module. By using programmatic navigation with history management, the system prevents users from accidentally navigating back into an empty state, ensuring the integrity of the testing session. The application is divided into four primary segments—Listening, Reading, Writing, and a final Summary—each mapped to a specific URL path to maintain a clear and organized single-page structure.

By leveraging a component-based routing approach paired with TypeScript, the system achieves high type safety and prevents common navigation errors. This ensures that as a student progresses from listening comprehension through to the final writing assessment, the transition is fluid and the underlying data remains synchronized. The final summary interface serves as a comprehensive conclusion to the workflow, pulling all stored state together to provide a clear overview of the completed examination.

```
● ● ●
```

```
1  const Listening: React.FC = () => {
2    const schema = listeningData as any;
3    const questions = schema.questions as any[];
4    const audioSrc = schema.audioSrc as string;
5    const [current, setCurrent] = useState(1);
6    const { listeningAnswers, saveListeningAnswer } = useExam();
7
8    const total = questions.length;
9    const currentQuestion = useMemo(() => questions[current - 1] || null, [questions, current]);
10
11   const handleAnswer = (qId: string, answer: string) => {
12     saveListeningAnswer(qId, answer);
13   };
14
15   return (
16     <div className="min-h-screen flex items-start justify-center bg-gray-50 py-8">
17       <div className="w-full max-w-3xl px-4">
18         <div className="mb-6 bg-white p-4 rounded-lg shadow">
19           <div className="font-bold mb-2">Listening Section</div>
20           <audio controls className="w-full">
21             <source src={audioSrc} />
22             Your browser does not support the audio element.
23           </audio>
24         </div>
25
26         {currentQuestion && (
27           <QuestionCard
28             question={currentQuestion}
29             selected={listeningAnswers[currentQuestion.id]}
30             onAnswer={handleAnswer}
31           />
32         )}
33
34         <Pagination
35           currentPage={current}
36           totalPages={Math.max(1, total)}
37           onPrev={() => setCurrent((p) => Math.max(1, p - 1))}
38           onNext={() => setCurrent((p) => Math.min(total, p + 1))}
39           setPage={(p) => setCurrent(p)}
40         />
41       </div>
42     </div>
43   );
44 };
45
46 export default Listening;
47
```

Figure 12: Code Snippet of Online Exam Site Routing

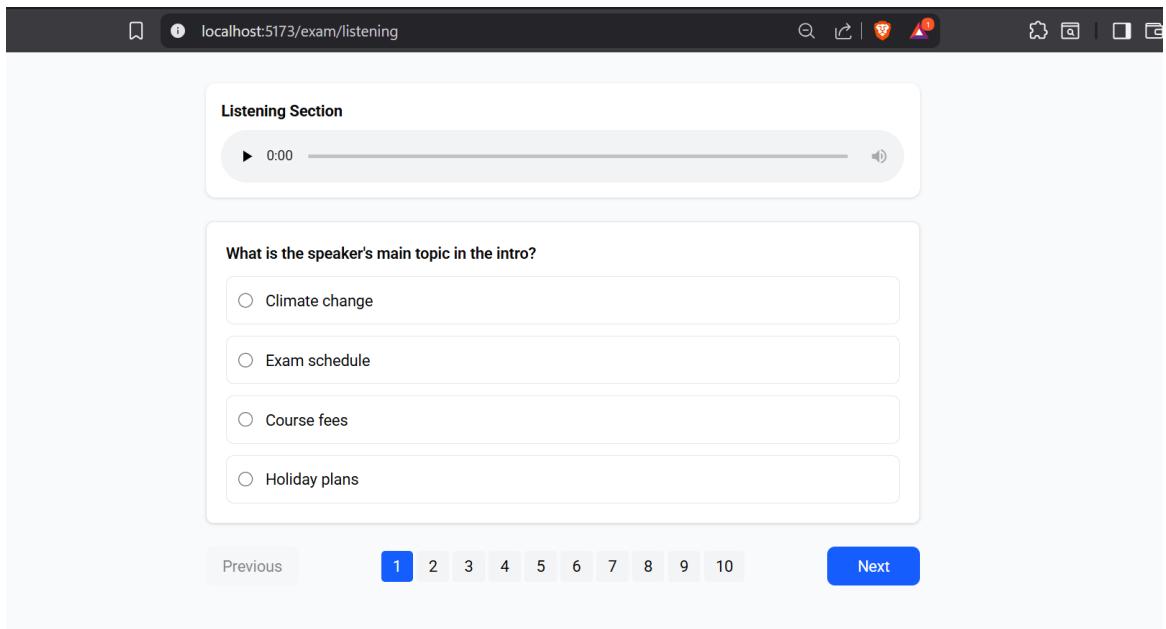


Figure 13: Online Exam Site Listening Section

```
 1  const Writing: React.FC = () => {
 2    const { writingAnswers, saveWriting } = useExam();
 3    const [local1, setLocal1] = useState(writingAnswers.task1 || "");
 4    const [local2, setLocal2] = useState(writingAnswers.task2 || "");
 5    const words1 = useMemo(() => countWords(local1), [local1]);
 6    const words2 = useMemo(() => countWords(local2), [local2]);
 7
 8    const onSave = (task: "task1" | "task2", text: string) => {
 9      saveWriting(task, text);
10    };
11
12    return (
13      <div className="min-h-screen flex items-start justify-center bg-gray-50 py-8">
14        <div className="w-full max-w-3xl px-4">
15          <div className="bg-white p-6 rounded-lg shadow flex-col gap-6">
16            <div>
17              <div className="font-bold text-xl mb-2">Writing Task 1</div>
18              <div className="text-sm text-gray-600 mb-2">Write at least {MIN_TASK1} words.</div>
19              <textarea
20                value={local1}
21                onChange={(e) => {
22                  setLocal1(e.target.value);
23                  onSave("task1", e.target.value);
24                }}
25                rows={6}
26                className="w-full border rounded-md p-3 focus:ring-2 focus:ring-blue-400"
27              />
28              <div className="flex justify-between text-sm mt-2">
29                <div>{words1} words</div>
30                <div className={ font-medium ${words1 < MIN_TASK1 ? "text-red-600" : "text-green-600"} }>
31                  {words1 < MIN_TASK1 ? `Minimum ${MIN_TASK1}` : "OK"}
32                </div>
33              </div>
34            </div>
35
36            <div>
37              <div className="font-bold text-xl mb-2">Writing Task 2</div>
38              <div className="text-sm text-gray-600 mb-2">Write at least {MIN_TASK2} words.</div>
39              <textarea
40                value={local2}
41                onChange={(e) => {
42                  setLocal2(e.target.value);
43                  onSave("task2", e.target.value);
44                }}
45                rows={10}
46                className="w-full border rounded-md p-3 focus:ring-2 focus:ring-blue-400"
47              />
48              <div className="flex justify-between text-sm mt-2">
49                <div>{words2} words</div>
50                <div className={ font-medium ${words2 < MIN_TASK2 ? "text-red-600" : "text-green-600"} }>
51                  {words2 < MIN_TASK2 ? `Minimum ${MIN_TASK2}` : "OK"}
52                </div>
53              </div>
54            </div>
55          </div>
56        </div>
57      </div>
58    );
59  };
60
61  export default Writing;
62
```

Figure 14: Online Exam Site Speaking Section

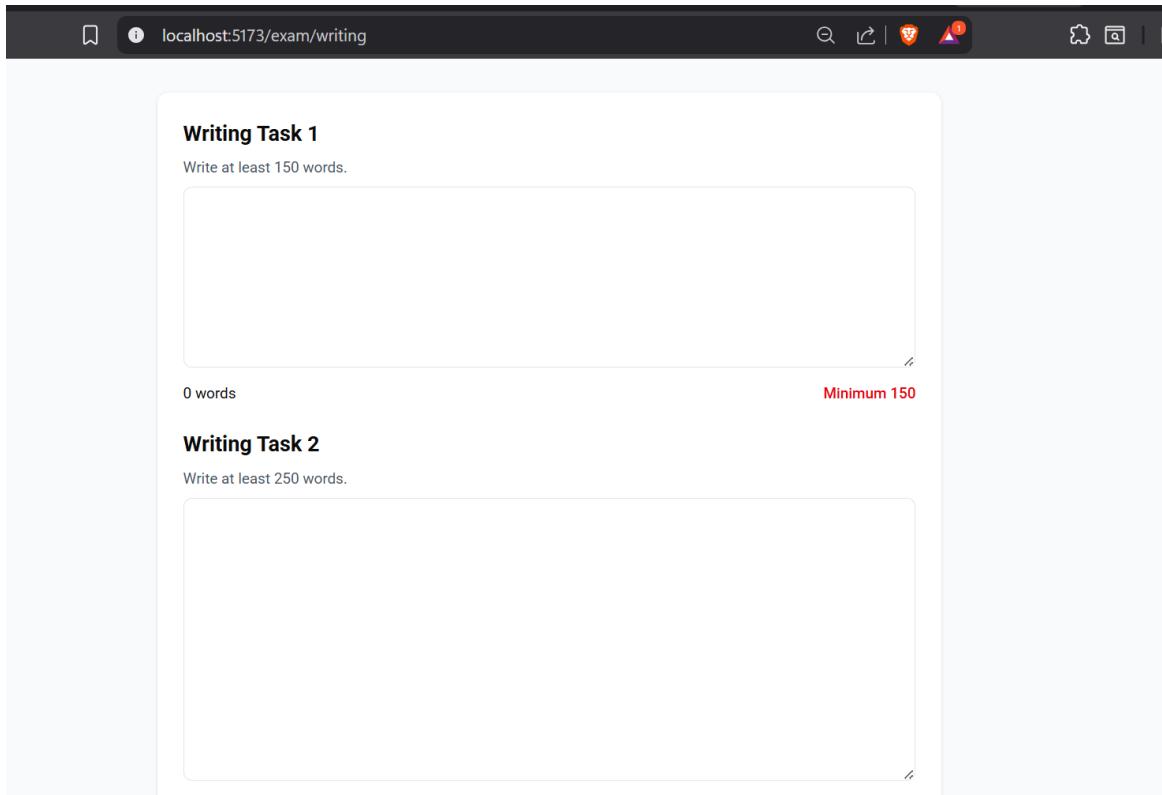


Figure 15: Online Exam Site Writing Section

```
● ○ ●
1 const Reading: React.FC = () => {
2   const schema = readingData as any;
3   const questions = schema.questions as any[];
4   const passage = schema.passage as string;
5   const [current, setCurrent] = useState(1);
6   const { readingAnswers, saveReadingAnswer } = useExam();
7
8   const total = questions.length;
9   const currentQuestion = useMemo(() => questions[current - 1] || null, [questions, current]);
10
11  const handleAnswer = (qId: string, answer: string) => {
12    saveReadingAnswer(qId, answer);
13  };
14
15  return (
16    <div className="min-h-screen flex items-start justify-center bg-gray-50 py-8">
17      <div className="w-full max-w-3xl px-4">
18        <div className="mb-6 bg-white p-6 rounded-lg shadow">
19          <div className="font-bold mb-3">Reading Passage</div>
20          <p className="text-gray-700 leading-relaxed">{passage || "Loading passage..."}</p>
21        </div>
22
23        {currentQuestion && (
24          <QuestionCard
25            question={currentQuestion}
26            selected={readingAnswers[currentQuestion.id]}
27            onAnswer={handleAnswer}
28          />
29        )}
30
31        <Pagination
32          currentPage={current}
33          totalPages={Math.max(1, total)}
34          onPrev={() => setCurrent((p) => Math.max(1, p - 1))}
35          onNext={() => setCurrent((p) => Math.min(total, p + 1))}
36          setPage={(p) => setCurrent(p)}
37          />
38        </div>
39      </div>
40    );
41  );
42
43  export default Reading;
44
```

Figure 16: Code Snippet of Online Exam Site Summary Section

A screenshot of a web browser displaying an online exam reading section. The URL in the address bar is `localhost:5173/exam/reading`. The page title is "Reading Passage". The main content area contains a passage about climate change affecting daily life, mentioning agriculture, city planning, and energy usage. Below the passage is a question: "Which area is mentioned as adapting to climate change?" with four options: Sports, Agriculture, Fashion, and Entertainment. At the bottom, there is a navigation bar with "Previous" and "Next" buttons, and a page number "1" highlighted among numbered buttons 1 through 10.

Reading Passage

Climate change affects many aspects of daily life. The passage below explores how communities adapt to changing seasons and weather patterns. It covers strategies in agriculture, city planning, and energy usage. Readers should pay attention to examples and reasoning used to support each claim.

Which area is mentioned as adapting to climate change?

Sports

Agriculture

Fashion

Entertainment

Previous 1 2 3 4 5 6 7 8 9 10 Next

Figure 17: Online Exam Site Reading Section

4. Square Japan Landing Page

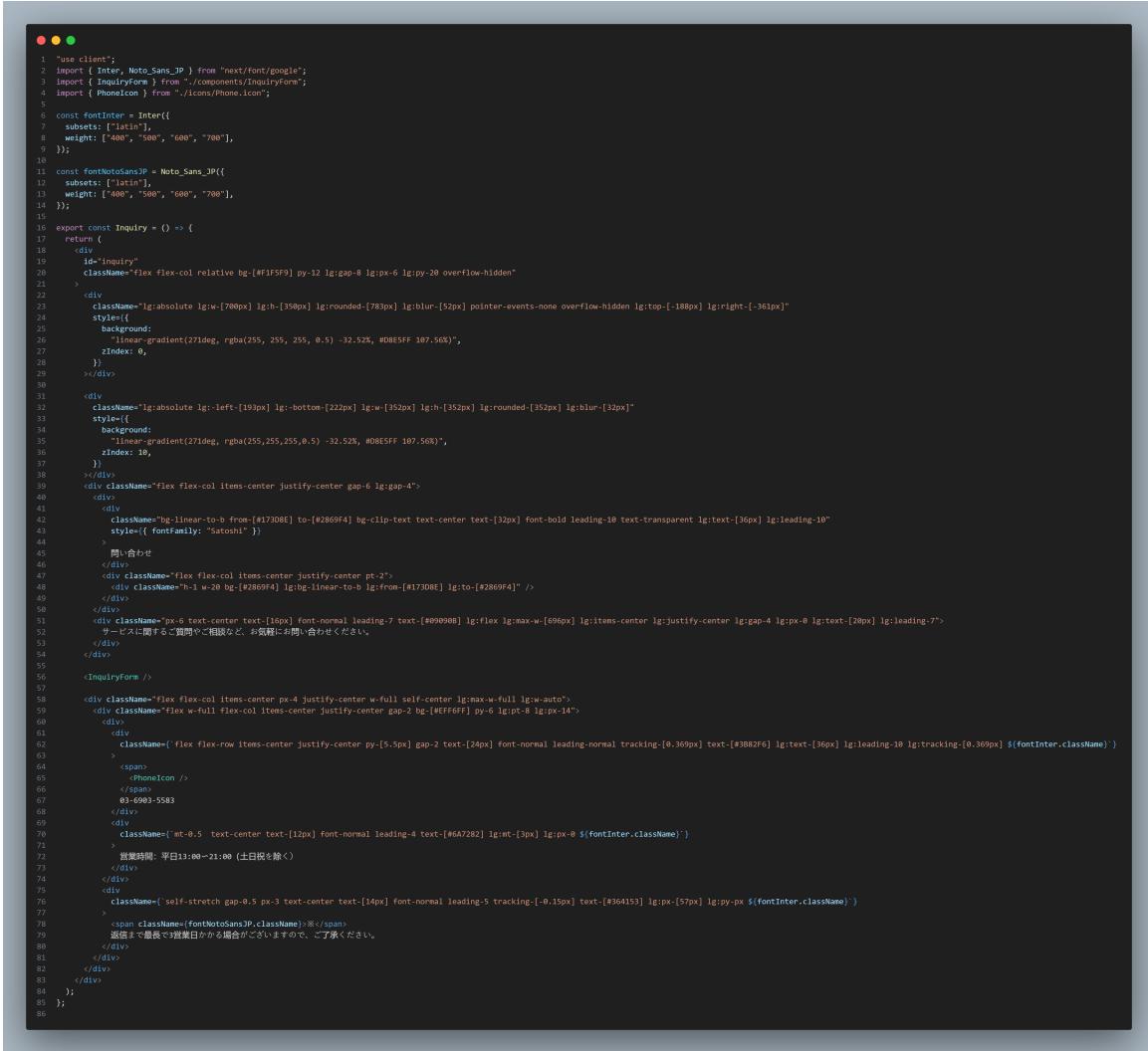
The Square Japan landing page was developed using Next.js, meticulously translating a Figma design into a high-performance web experience. To increase user engagement, the header section incorporates smooth, timed animations that activate upon page load or as the user scrolls, creating a polished and interactive first impression. These transitions provide a modern feel while guiding the user's attention naturally toward the primary messaging of the site.

Information architecture was a key focus, particularly within the FAQ section. Utilizing Shadcn UI components, this area features an accordion-based layout to keep the interface clean while providing immediate answers to common questions. To further improve usability, a tabbed navigation system was integrated, allowing users to switch between different FAQ categories effortlessly. This ensures that even large volumes of information remain organized and accessible without cluttering the landing page.

The communication and recruitment workflows were designed for maximum reliability and brand consistency. A dedicated modal dialog was implemented for the “Join Our Team” feature, offering a streamlined path for applicants to submit their details. Behind the scenes, custom email templates were developed for both general inquiries and recruitment forms to ensure all outreach reflects the professional branding of the project. These workflows are powered by the Resend email service, which automates confirmation messages to provide users with immediate and reliable feedback upon every submission.

```
1  /*use client*/
2  import { Inter } from "next/font/google";
3  import { InquiryForm } from "./components/InquiryForm";
4  import { PhoneIcon } from "./icons/Phone.icon";
5
6  const fontInter = Inter({
7    subsets: ["latin"],
8    weight: ["400", "500", "600", "700"],
9  });
10
11 const fontNotoSansJP = Noto_Sans_JP({
12   subsets: ["latin"],
13   weight: ["400", "500", "600", "700"],
14 });
15
16 export const Inquiry = () => {
17   return (
18     <div id="inquiry">
19       <InquiryForm />
20     </div>
21   );
22 }
23
24 .className="flex flex-col relative bg-[#F1F5F9] py-12 lg:gap-8 lg:px-6 lg:py-20 overflow-hidden"
25 >
26   <div style={background: "linear-gradient(271deg, rgba(255, 255, 255, 0.5) -32.52%, #0BE5FF 107.56%)", zIndex: 0}>
27     <div>
28       <div style={background: "linear-gradient(271deg, rgba(255, 255, 255, 0.5) -32.52%, #0BE5FF 107.56%)", zIndex: 10}>
29         <div>
30           <div style={fontFamily: "Satoshi"}>
31             開い含わせ
32           </div>
33           <div>
34             <h1>w-20 bg-[#2869F4]</h1>
35             <div>
36               <p>お問い合わせに関するご質問やご相談など、お気軽に問い合わせください。
37             </div>
38           </div>
39         </div>
40       </div>
41     </div>
42   </div>
43   <div style={fontFamily: "Satoshi"}>
44     <div>
45       <span>お問い合わせ</span>
46       <span>お問い合わせ</span>
47       <span>お問い合わせ</span>
48       <span>お問い合わせ</span>
49       <span>お問い合わせ</span>
50       <span>お問い合わせ</span>
51       <span>お問い合わせ</span>
52       <span>お問い合わせ</span>
53       <span>お問い合わせ</span>
54     </div>
55   </div>
56 </InquiryForm />
57
58 <div>
59   <div>
60     <div>
61       <div>
62         <div>
63           <span>お問い合わせ</span>
64           <span>お問い合わせ</span>
65           <span>お問い合わせ</span>
66           <span>お問い合わせ</span>
67           <span>お問い合わせ</span>
68           <span>お問い合わせ</span>
69           <span>お問い合わせ</span>
70           <span>お問い合わせ</span>
71           <span>お問い合わせ</span>
72           <span>お問い合わせ</span>
73           <span>お問い合わせ</span>
74           <span>お問い合わせ</span>
75           <span>お問い合わせ</span>
76           <span>お問い合わせ</span>
77           <span>お問い合わせ</span>
78           <span>お問い合わせ</span>
79           <span>お問い合わせ</span>
80           <span>お問い合わせ</span>
81           <span>お問い合わせ</span>
82           <span>お問い合わせ</span>
83           <span>お問い合わせ</span>
84         </div>
85       </div>
86     </div>
87   </div>
88 </div>
89 </div>
90 </div>
91 </div>
92 </div>
93 </div>
94 </div>
95 </div>
96 </div>
97 </div>
98 </div>
99 </div>
100 </div>
101 </div>
102 </div>
103 </div>
104 </div>
105 </div>
106 </div>
107 </div>
108 </div>
109 </div>
110 </div>
111 </div>
112 </div>
113 </div>
114 </div>
115 </div>
116 </div>
117 </div>
118 </div>
119 </div>
120 </div>
121 </div>
122 </div>
123 </div>
124 </div>
125 </div>
126 </div>
127 </div>
128 </div>
129 </div>
130 </div>
131 </div>
132 </div>
133 </div>
134 </div>
135 </div>
136 </div>
137 </div>
138 </div>
139 </div>
140 </div>
141 </div>
142 </div>
143 </div>
144 </div>
145 </div>
146 </div>
147 </div>
148 </div>
149 </div>
150 </div>
151 </div>
152 </div>
153 </div>
154 </div>
155 </div>
156 </div>
157 </div>
158 </div>
159 </div>
160 </div>
161 </div>
162 </div>
163 </div>
164 </div>
165 </div>
166 </div>
167 </div>
168 </div>
169 </div>
170 </div>
171 </div>
172 </div>
173 </div>
174 </div>
175 </div>
176 </div>
177 </div>
178 </div>
179 </div>
180 </div>
181 </div>
182 </div>
183 </div>
184 </div>
185 </div>
186 </div>
```

Figure 18: Code Snippet of Square Japan inquiry Page



```

1  "use client";
2  import { Inter, Noto_Sans_JP } from "next/font/google";
3  import { InquiryForm } from "./components/inquiryForm";
4  import { PhoneIcon } from "./icons/Phone.icon";
5
6  const fontInter = Inter({
7    subsets: ['latin'],
8    weights: ['400', '500', '600', '700'],
9  });
10
11 const fontNotoSansJP = Noto_Sans_JP({
12   subsets: ['latin'],
13   weights: ['400', '500', '600', '700'],
14 });
15
16 export const Inquiry = () => {
17   return (
18     <div id="inquiry" style={{flex: 1, padding: 12, gap: 8, display: 'flex', flexWrap: 'wrap', overflow: 'hidden'}}>
19       <div style={{width: '100%', height: '100%', position: 'relative'}}>
20         <div style={{position: 'absolute', left: 0, top: 0, width: '100%', height: '100%', background: 'linear-gradient(270deg, rgba(255, 255, 255, 0.5) -32.5%, #00E5FF 107.5%)', zIndex: 0}}>
21         </div>
22         <div style={{position: 'absolute', left: 0, top: 0, width: '100%', height: '100%', background: 'linear-gradient(270deg, rgba(255, 255, 255, 0.5) -32.5%, #00E5FF 107.5%), linear-gradient(270deg, rgba(255, 255, 255, 0.5) -32.5%, #00E5FF 107.5%)', zIndex: 10}}>
23         </div>
24       </div>
25       <div style={{display: 'flex', flex: 1, gap: 4, align-items: 'center', justify-content: 'space-between'}}>
26         <div style={{flex: 1, gap: 2, display: 'flex', align-items: 'center'}}>
27           <div style={{flex: 1, gap: 2, display: 'flex', align-items: 'center'}}>
28             <div style={{flex: 1, gap: 2, display: 'flex', align-items: 'center'}}>
29               <div style={{flex: 1, gap: 2, display: 'flex', align-items: 'center'}}>
30                 <div style={{flex: 1, gap: 2, display: 'flex', align-items: 'center'}}>
31                   <div style={{flex: 1, gap: 2, display: 'flex', align-items: 'center'}}>
32                     <div style={{flex: 1, gap: 2, display: 'flex', align-items: 'center'}}>
33                       <div style={{flex: 1, gap: 2, display: 'flex', align-items: 'center'}}>
34                         <div style={{flex: 1, gap: 2, display: 'flex', align-items: 'center'}}>
35                           <div style={{flex: 1, gap: 2, display: 'flex', align-items: 'center'}}>
36                             <div style={{flex: 1, gap: 2, display: 'flex', align-items: 'center'}}>
37                               <div style={{flex: 1, gap: 2, display: 'flex', align-items: 'center'}}>
38                                 <div style={{flex: 1, gap: 2, display: 'flex', align-items: 'center'}}>
39                                   <div style={{flex: 1, gap: 2, display: 'flex', align-items: 'center'}}>
40                                     <div style={{flex: 1, gap: 2, display: 'flex', align-items: 'center'}}>
41                                       <div style={{flex: 1, gap: 2, display: 'flex', align-items: 'center'}}>
42                                         <div style={{flex: 1, gap: 2, display: 'flex', align-items: 'center'}}>
43                                           <div style={{flex: 1, gap: 2, display: 'flex', align-items: 'center'}}>
44                                             <div style={{flex: 1, gap: 2, display: 'flex', align-items: 'center'}}>
45                                               <div style={{flex: 1, gap: 2, display: 'flex', align-items: 'center'}}>
46                                                 <div style={{flex: 1, gap: 2, display: 'flex', align-items: 'center'}}>
47                                                   <div style={{flex: 1, gap: 2, display: 'flex', align-items: 'center'}}>
48                                                     <div style={{flex: 1, gap: 2, display: 'flex', align-items: 'center'}}>
49                                                       <div style={{flex: 1, gap: 2, display: 'flex', align-items: 'center'}}>
50                                                         <div style={{flex: 1, gap: 2, display: 'flex', align-items: 'center'}}>
51               サービスに関するご質問やご相談など、お気軽にお問い合わせください。
52             </div>
53           </div>
54         </div>
55       </div>
56     </InquiryForm />
57   </div>
58   <div style={{flex: 1, gap: 2, display: 'flex', align-items: 'center'}}>
59     <div style={{flex: 1, gap: 2, display: 'flex', align-items: 'center'}}>
60       <div style={{flex: 1, gap: 2, display: 'flex', align-items: 'center'}}>
61         <div style={{flex: 1, gap: 2, display: 'flex', align-items: 'center'}}>
62           <div style={{flex: 1, gap: 2, display: 'flex', align-items: 'center'}}>
63             <div style={{flex: 1, gap: 2, display: 'flex', align-items: 'center'}}>
64               <span style={{fontFamily: 'Satoshi', color: '#00008B', fontSize: 14px}}>
65                 <PhoneIcon />
66               </span>
67             </div>
68           </div>
69         </div>
70         <div style={{flex: 1, gap: 2, display: 'flex', align-items: 'center'}}>
71           <div style={{flex: 1, gap: 2, display: 'flex', align-items: 'center'}}>
72             <div style={{flex: 1, gap: 2, display: 'flex', align-items: 'center'}}>
73               <div style={{flex: 1, gap: 2, display: 'flex', align-items: 'center'}}>
74                 <div style={{flex: 1, gap: 2, display: 'flex', align-items: 'center'}}>
75                   <div style={{flex: 1, gap: 2, display: 'flex', align-items: 'center'}}>
76                     <div style={{flex: 1, gap: 2, display: 'flex', align-items: 'center'}}>
77                       <div style={{flex: 1, gap: 2, display: 'flex', align-items: 'center'}}>
78                         <span style={{fontWeight: 'bold', color: '#00008B', fontSize: 14px}}>
79                           選択まで最終でご選択日がかかる場合がございますので、ご了承ください。
80                         </span>
81                       </div>
82                     </div>
83                   </div>
84                 </div>
85               </div>
86             </div>
87           </div>
88         </div>
89       </div>
90     </div>
91   </div>
92 
```

Figure 19: Code snippet of the Square Japan inquiry page (full view)

問い合わせ

サービスに関するご質問やご相談など、お気軽にお問い合わせください。

お名前 <small>必須</small>	山田 太郎
会社名 <small>必須</small>	株式会社□□
メールアドレス <small>必須</small>	example@hoslog.work
電話番号 <small>必須</small>	09012345678
お問い合わせ内容 <small>必須</small>	お問い合わせ内容を記入ください（10文字以上、500文字以内）

送信する

📞 03-6903-5583
営業時間：平日13:00～21:00（土日祝を除く）
※返信まで最長で3営業日かかる場合がございますので、ご了承ください。

Figure 20: Code Snippet of Square Japan Landing Page

4. Tapaiko Bazar E-commerce Website

Tapaiko Bazar project involved building a comprehensive, web-based dashboard system from the ground up. Developed through a close partnership between frontend and backend teams, the platform features distinct interfaces for administrators and clients. Our goal was to create a secure, scalable, and user-friendly ecosystem that simplifies vendor and product management while ensuring a smooth experience for everyday users.

The administrative side features a structured authentication system designed for security and ease of use. A clean, card-based login interface utilizes Zod for schema-based validation, ensuring that email and password inputs are accurate before submission.

To manage credential recovery, a modular "Forgot Password" workflow was implemented. This process follows a three-step sequence—email submission, one-time password (OTP) verification, and a secure password reset. Each step is handled through dedicated API endpoints, maintaining a clear separation between frontend form handling and backend security logic. Efficient vendor and product management are central to the administrative workflow. The vendor management module utilizes a responsive data table and Tailwind CSS to display records clearly across all devices. Similarly, the product management interface employs a two-column layout that separates product metadata from media management. By using the FormData API for image uploads and implementing automatic input resets upon successful submission, the system maintains a clean and high-performance environment for handling complex product data and multi-attribute variations.

On the client-facing side, the platform emphasizes intuitive navigation and discovery. A dynamic filtering system allows for product searches based on tags and price ranges, with states synchronized to URL parameters to ensure persistence during navigation. To further enhance the shopping experience, an intelligent variant selection algorithm evaluates product combinations in real-time, preventing users from selecting unavailable options. This is complemented by a custom carousel wrapper built for smooth image galleries and a dedicated website review module that displays overall platform feedback through dynamic star ratings and optimized imagery.

The platform maintains high standards of consistency by mirroring security workflows across all user roles. The password recovery process for clients utilizes the same multi-step modal design as the administrative side, providing a familiar and secure method for regaining account access. By integrating reusable UI components and synchronized backend services, the system ensures real-time data accuracy and a cohesive user experience throughout the entire application.



```
1 "use client";
2
3 import { Label } from "@ui/label";
4 import { LoginFormSchema, type LoginFormType } from "./AdminLoginForm.zod";
5 import { zodResolver } from "@hookform/resolvers/zod";
6 import { useForm } from "react-hook-form";
7 import { Button } from "@ui/button";
8 import { useLogin } from "@hooks/auth/useAuth";
9 import { Input } from "../Input/Input";
10 import { Dialog } from "../Dialog/Dialog";
11 import { ForgotPassword } from "../ForgotPassword/ForgotPassword";
12 import { useState } from "react";
13
14 export const AdminLoginForm = () => {
15   const { mutate, isPending } = useLogin();
16   const {
17     register,
18     handleSubmit,
19     reset,
20
21     formState: { errors },
22   } = useForm<LoginFormType>({
23     resolver: zodResolver(LoginFormSchema),
24     mode: "onChange",
25   });
26   const [open, setOpen] = useState(false);
27   const onSubmit = (data: LoginFormType) => {
28     reset();
29     mutate(data);
30   };
31
32   return (
33     <form
34       onSubmit={handleSubmit(onSubmit)}
35       className="py-6 flex flex-col gap-2"
36     >
37       <div className="text-3xl font-semibold mb-5">Login</div>
38       <div className="flex flex-col gap-3">
39         <Label htmlFor="email">Email</Label>
40         <Input
41           id="email"
42           {...register("email")}
43           type="email"
44           placeholder="Enter your email"
45         />
46         <p className="text-red-500">{errors.email?.message}</p>
47       </div>
48       <div className="flex flex-col gap-3">
49         <Label htmlFor="password">Password</Label>
50         <Input
51           {...register("password")}
52           type="password"
53           placeholder="Enter your password"
54         />
55         <p className="text-red-500">{errors.password?.message}</p>
56       </div>
57       <div>
58         <Button type="submit" className="mt-5 w-full" value="Login">
59           {isPending ? "Logging in..." : "Login"}
60         </Button>
61       </div>
62       <Dialog
63         open={open}
64         onChange={setOpen}
65         triggerContent={
66           <button className="text-sm underline" onClick={() => setOpen(true)}>
67             Forgot Password
68           </button>
69         }
70       >
71         <ForgotPassword onClose={() => setOpen(false)} />
72       </Dialog>
73     </form>
74   );
75 };
76
```

Figure 21: Tapaiko Bazar Admin Login Interface



```

1 import { useForm, type SubmitHandler } from "react-hook-form";
2 import { zodResolver } from "zod";
3 import {
4   AddVendorFormSchema,
5   type AddVendorFormValues,
6 } from "./AddVendorFormSchema.zod.ts";
7 import { useCreateVendor } from "@hooks/vendor/useCreateVendor.ts";
8 import { Button } from "@ui/button.tsx";
9 export const AddVendorForm = () => {
10   const { mutate } = useCreateVendor();
11   const {
12     register,
13     handleSubmit,
14     reset,
15     formState: { errors },
16   } = useForm<AddVendorFormValues>({
17     resolver: zodResolver(AddVendorFormSchema),
18     defaultValues: {
19       name: "",
20       contactPerson: "",
21       email: "",
22       phone: "",
23       address: "",
24     },
25     mode: "onChange",
26   });
27
28   const onSubmit: SubmitHandler<AddVendorFormValues> = (
29     data: AddVendorFormValues,
30   ) => {
31     mutate({
32       name: data.name,
33       contactPerson: data.contactPerson,
34       email: data.email,
35       phone: data.phone,
36       address: data.address,
37     });
38     reset();
39   };
40
41   return (
42     <form onSubmit={handleSubmit(onSubmit)}>
43       <div className="flex flex-col gap-4">
44         <div>
45           <label>Company Name</label>
46           <input
47             type="text"
48             placeholder="Enter company name"
49             className="mt-1 block w-full border border-gray-300 rounded-md shadow-sm p-2"
50             {...register("name")}>
51           </input>
52           {errors.name && (
53             <p className="text-sm text-red-500 mt-1">{errors.name.message}</p>
54           )}
55         </div>
56
57         <div>
58           <label>Contact Person</label>
59           <input
60             type="text"
61             placeholder="Enter contact person"
62             className="mt-1 block w-full border border-gray-300 rounded-md shadow-sm p-2"
63             {...register("contactPerson")}>
64           </input>
65           {errors.contactPerson && (
66             <p className="text-sm text-red-500 mt-1">{errors.contactPerson.message}</p>
67           )}
68         </div>
69
70         <div>
71           <label>Email</label>
72           <input
73             type="email"
74             placeholder="Enter email"
75             className="mt-1 block w-full border border-gray-300 rounded-md shadow-sm p-2"
76             {...register("email")}>
77           </input>
78           {errors.email && (
79             <p className="text-sm text-red-500 mt-1">{errors.email.message}</p>
80           )}
81         </div>
82
83         <div>
84           <label>Phone</label>
85           <input
86             type="tel"
87             placeholder="Enter phone number"
88             className="mt-1 block w-full border border-gray-300 rounded-md shadow-sm p-2"
89             {...register("phone")}>
90           </input>
91           {errors.phone && (
92             <p className="text-sm text-red-500 mt-1">{errors.phone.message}</p>
93           )}
94         </div>
95
96         <div>
97           <label>Address</label>
98           <input
99             type="text"
100             placeholder="Enter address"
101             className="mt-1 block w-full border border-gray-300 rounded-md shadow-sm p-2"
102             {...register("address")}>
103           </input>
104           {errors.address && (
105             <p className="text-sm text-red-500 mt-1">{errors.address.message}</p>
106           )}
107         </div>
108
109       <div className="flex justify-center mt-4">
110         <Button
111           type="submit"
112           variant="default"
113           className="px-4 py-4 text-white rounded-md">
114           Add Vendor
115         </Button>
116       </div>
117     </form>
118   );
119 }

```

Figure 22: Tapaiko Bazar Vendor Form Module

Vendors

Vendors Table

All Vendors Active Vendors Inactive Vendors

Vendor Id	Company Name	Contact Person	Email	Phone	Address	Joined On	Status	Actions
1	Samsung Nepal	Utsarga	utsarga@gmail.com	9801122334	Kalimati	2026-01-09	• Active	
2	Local Books Supplier	Amar	limbuamardeep18@gmail.co	9821313412	Kapan	2026-01-09	● Inactive	
3	Blienawf	Claref	contact@claref.com	9801122334	string	2026-01-09	• Active	
4	Brand Nepal	Bikin Nepal	bikinnepal@gmail.com	9843522535	Kapan	2026-01-09	• Active	

< Previous 1 Next >

Figure 23: Vendor Table Overview

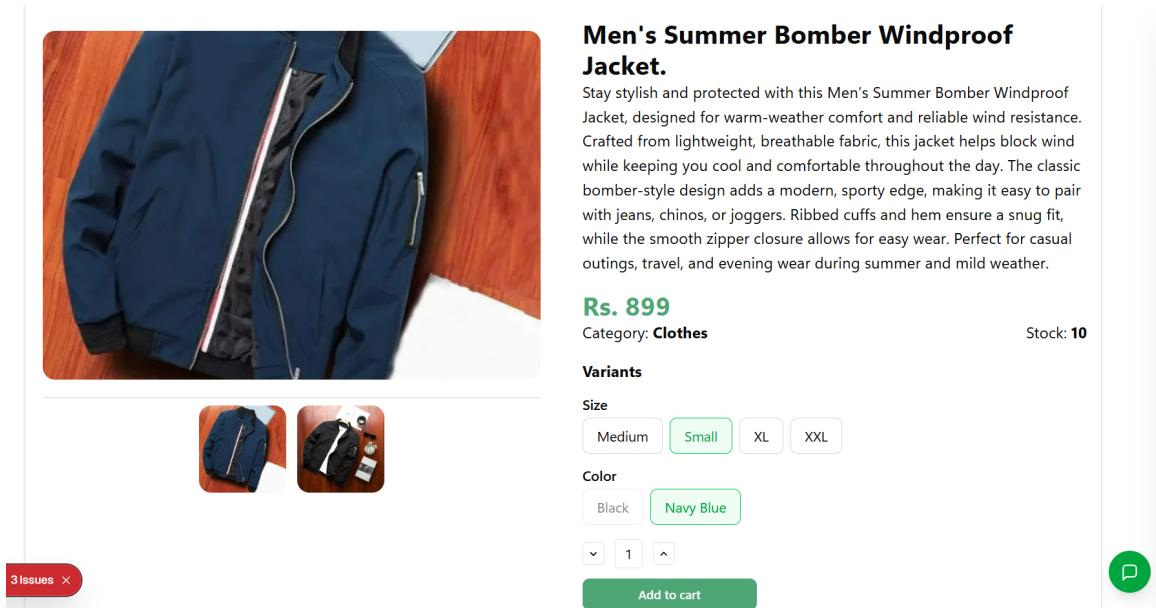


Figure 24: Product Details and Management View

Product Management



Add New Product

Publish Product

Basic Details

Product Name

Short Description

Description

Highlight this product in featured section

Upload Product Details

Upload Image

Categories

Product Categories

Attributes

Color Size RAM Weight
 Storage Type Capacity Shoes size
 Lens Quantity

Publish Product

Figure 25: Product Management Dashboard

The screenshot shows a web browser displaying the eGujaratiBazar website at localhost:3000/product?tags=4K+Camera%2CBest+Seller. The page features a navigation bar with Home, Product (underlined), About Us, and Contact. On the left, there's a sidebar with 'Categories' (All, Grocery, Clothes, Shoes, Electronics) and a 'Filter' section (Tags: 4K Camera, Best Seller, Electronics, Fashion). A red button indicates '5 Issues'. The main content area displays four products: Canon Camera (Rs. 3000), LG Refrigerator (Rs. 89000), Reebok Nano X (Rs. 13000), and Puma Smash v2 (Rs. 10000). Each product card includes a 'View Details' button and an 'Add to cart' button.

Figure 26: Filtered Product Data View

34

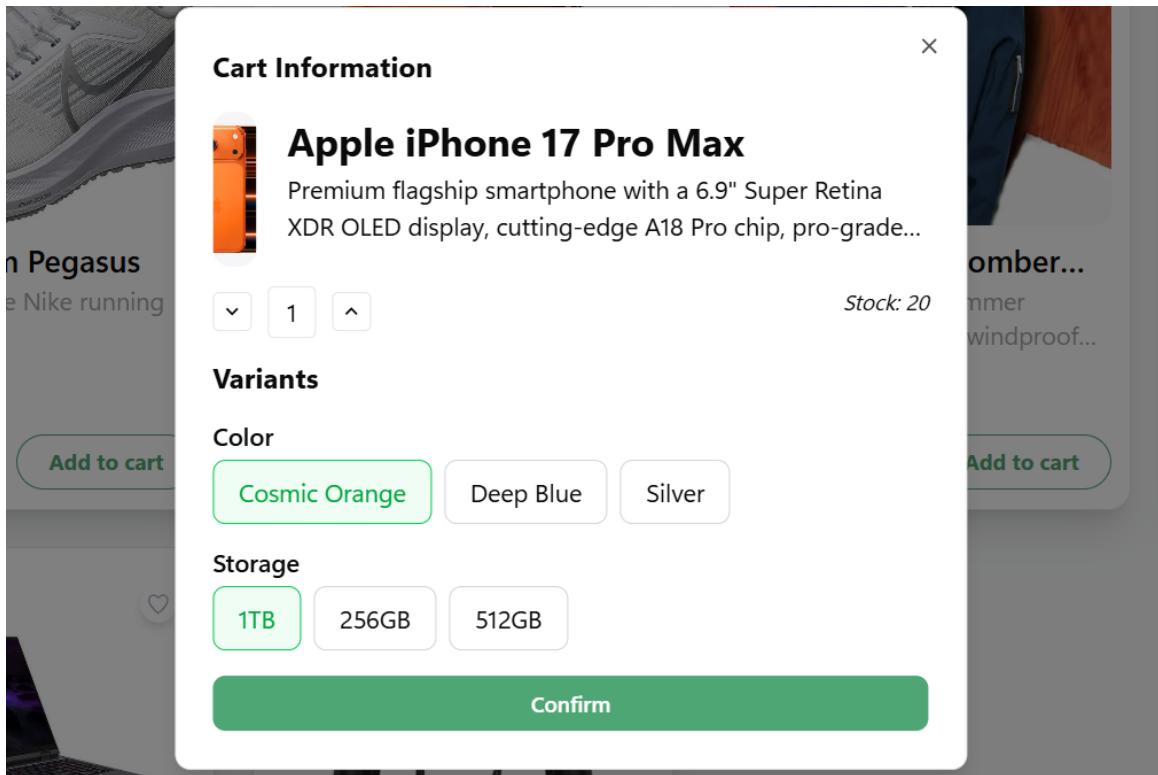


Figure 27: Client Portal Product Cart Interface

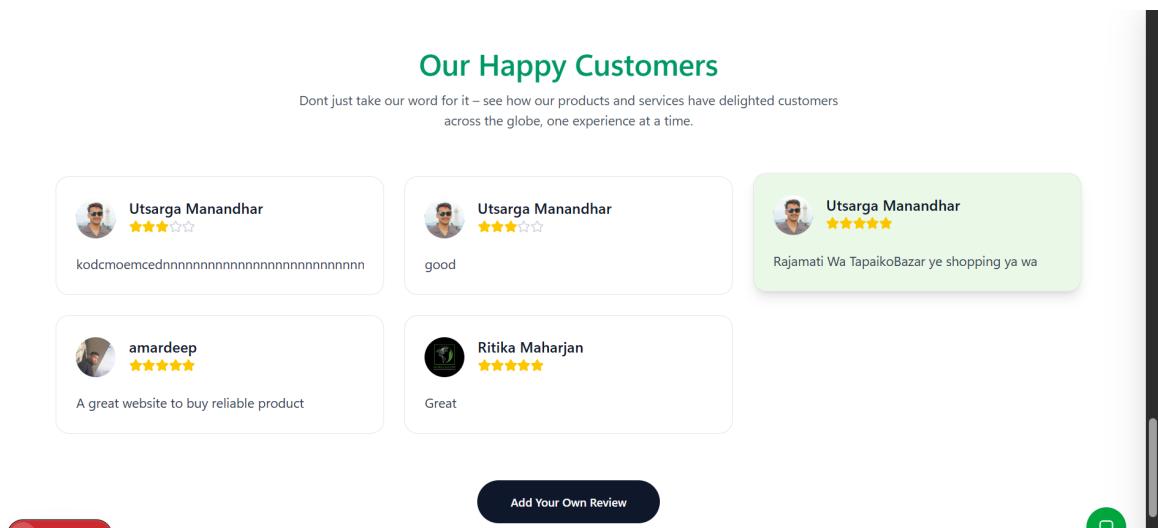
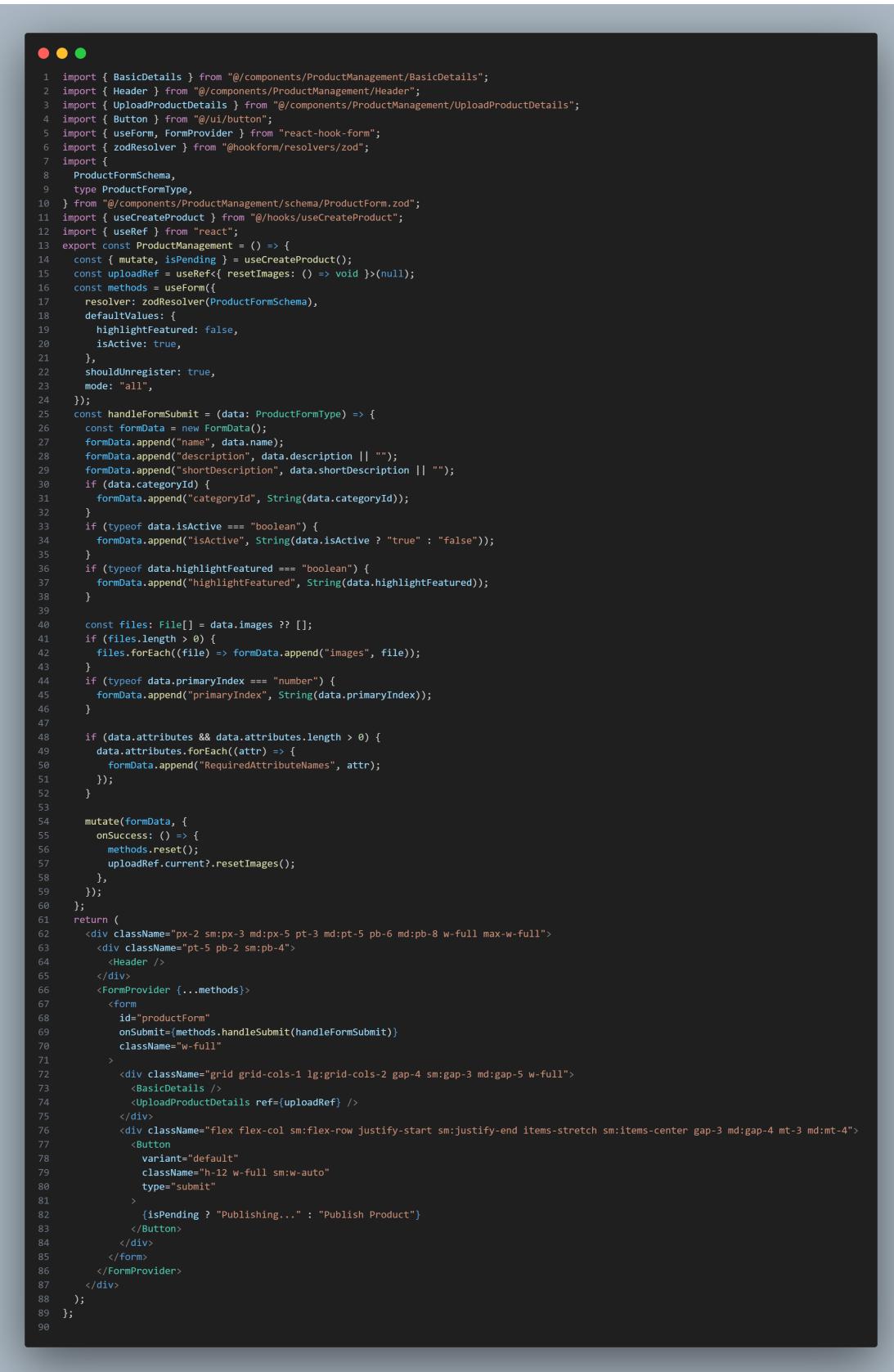


Figure 28: Website Review and Rating Component



```

1 import { BasicDetails } from "@/components/ProductManagement/BasicDetails";
2 import { Header } from "@/components/ProductManagement/Header";
3 import { UploadProductDetails } from "@/components/ProductManagement/UploadProductDetails";
4 import { Button } from "@ui/button";
5 import { useForm, FormProvider } from "react-hook-form";
6 import { zodResolver } from "@hookform/resolvers/zod";
7 import {
8   ProductFormSchema,
9   type ProductFormType,
10 } from "@/components/ProductManagement/schema/ProductForm.zod";
11 import { useCreateProduct } from "@/hooks/useCreateProduct";
12 import { useRef } from "react";
13 export const ProductManagement = () => {
14   const { mutate, isPending } = useCreateProduct();
15   const uploadRef = useRef({ resetImages: () => void 0 }>(null));
16   const methods = useForm({
17     resolver: zodResolver(ProductFormSchema),
18     defaultValues: {
19       highlightFeatured: false,
20       isActive: true,
21     },
22     shouldUnregister: true,
23     mode: "all",
24   });
25   const handleFormSubmit = (data: ProductFormType) => {
26     const formData = new FormData();
27     formData.append("name", data.name);
28     formData.append("description", data.description || "");
29     formData.append("shortDescription", data.shortDescription || "");
30     if (data.categoryId) {
31       formData.append("categoryId", String(data.categoryId));
32     }
33     if (typeof data.isActive === "boolean") {
34       formData.append("isActive", String(data.isActive ? "true" : "false"));
35     }
36     if (typeof data.highlightFeatured === "boolean") {
37       formData.append("highlightFeatured", String(data.highlightFeatured));
38     }
39     const files: File[] = data.images ?? [];
40     if (files.length > 0) {
41       files.forEach((file) => formData.append("images", file));
42     }
43     if (typeof data.primaryIndex === "number") {
44       formData.append("primaryIndex", String(data.primaryIndex));
45     }
46   }
47   if (data.attributes && data.attributes.length > 0) {
48     data.attributes.forEach((attr) => {
49       formData.append("RequiredAttributeNames", attr);
50     });
51   }
52 }
53
54 mutate(formData, {
55   onSuccess: () => {
56     methods.reset();
57     uploadRef.current?.resetImages();
58   },
59 });
60 };
61 return (
62   <div className="px-2 sm:px-3 md:px-5 pt-3 md:pt-5 pb-6 md:pb-8 w-full max-w-full">
63     <div className="pt-5 pb-2 sm:pb-4">
64       <Header />
65     </div>
66     <FormProvider {...methods}>
67       <form
68         id="productForm"
69         onSubmit={methods.handleSubmit(handleFormSubmit)}
70         className="w-full"
71       >
72         <div className="grid grid-cols-1 lg:grid-cols-2 gap-4 sm:gap-3 md:gap-5 w-full">
73           <BasicDetails />
74           <UploadProductDetails ref={uploadRef} />
75         </div>
76         <div className="flex flex-col sm:flex-row justify-start sm:justify-end items-stretch sm:items-center gap-3 md:gap-4 mt-3 md:mt-4">
77           <Button
78             variant="default"
79             className="h-12 w-full sm:w-auto"
80             type="submit"
81           >
82             {isPending ? "Publishing..." : "Publish Product"}
83           </Button>
84         </div>
85       </form>
86     </FormProvider>
87   </div>
88 );
89 };

```

Figure 29: Tapaiko Bazar Product Management UI

```

1  "use client";
2
3  import Autoplay from "embla-carousel-autoplay";
4  import { useMemo } from "react";
5  import { Slides } from "./constants/Slides";
6  import { Carousel } from "../Carousel/Carousel";
7  import Link from "next/link";
8
9  export const ProductSlider = () => {
10    const autoplay = useMemo(
11      () => Autoplay({ delay: 6000, stopOnInteraction: true }),
12      []
13    );
14
15    return (
16      <Carousel
17        rootClassName="relative w-full h-[400px] overflow-hidden"
18        previousClassName="absolute left-4 top-1/2 -translate-y-1/2 bg-white text-gray-700 p-2 rounded-full shadow hover:bg-gray-100"
19        nextClassName="absolute right-4 top-1/2 -translate-y-1/2 bg-white text-gray-700 p-2 rounded-full shadow hover:bg-gray-100"
20        opts={{ loop: true }}
21        plugins={[autoplay]}
22        items={Slides.map((slide) => (
23          <div key={slide.key} className="relative w-full h-[400px]">
24            <img
25              src={slide.src}
26              alt={slide.alt}
27              className="w-full h-full object-cover"
28            />
29
30            {slide.overlay && (
31              <div className="absolute inset-0 flex items-center px-12 bg-black/30">
32                <div className="ml-5 text-white max-w-md">
33                  <h2 className="text-4xl font-bold">{slide.overlay.title}</h2>
34                  <p className="mt-2">{slide.overlay.description}</p>
35
36                  <Link
37                    href={`/product?categoryId=${slide.overlay.categoryId}`}
38                    className="inline-block mt-4 bg-white text-black px-6 py-2 rounded"
39                  >
40                    {slide.overlay.buttonText}
41                  </Link>
42                </div>
43              </div>
44            )}
45          </div>
46        )));
47      />
48    );
49  };
50

```

Figure 30: Product Slider with Media Gallery

```

1  const [selectedVariants, setSelectedVariants] = useState<
2    Record<string, string | undefined>
3  >(() => getDefaultSelectedVariants(productItems.data?.variants));
4
5  if (
6    productItems.data?.variants &&
7    Object.keys(selectedVariants).length === 0
8  ) {
9    const defaults = getDefaultSelectedVariants(productItems.data.variants);
10   if (Object.keys(defaults).length > 0) {
11     setSelectedVariants(defaults);
12   }
13 }
14
15 const activeVariant = productItems.data?.variants?.find((variant) =>
16   Object.entries(selectedVariants).every(
17     ([key, value]) => variant.attributes[key] === value,
18   ),
19 );
20
21 const handleVariantChange = (attributeName: string, value: string) => {
22   const variants = productItems.data?.variants ?? [];
23   const nextSelection = { ...selectedVariants, [attributeName]: value };
24
25   Object.keys(nextSelection).forEach((key) => {
26     if (key === attributeName) return;
27
28     const valid = variants.find(
29       (v) =>
30         v.attributes[attributeName] === value &&
31         v.attributes[key] === nextSelection[key],
32     );
33
34     if (!valid) {
35       const fallback = variants.find(
36         (v) => v.attributes[attributeName] === value,
37       );
38       nextSelection[key] = fallback?.attributes[key];
39     }
40   });
41
42   setSelectedVariants(nextSelection);
43 };
44
45 const isAvailable = (
46   variants: Variant[] | undefined,
47   selected: Record<string, string | undefined>,
48   attrName: string,
49   value: string,
50 ) => {
51   const temp = { ...selected, [attrName]: value };
52   return variants?.some((v) =>
53     Object.entries(temp).every(
54       ([key, val]) => !val || v.attributes[key] === val,
55     ),
56   );
57 };

```

Figure 31: Variant Handling in Product Creation

CHAPTER 4:

CONCLUSION AND LEARNING OUTCOMES

4.1 Conclusion

The internship at Global Square IT Company presented a valuable opportunity to translate theoretical knowledge into practical application within a professional environment. This role focused on frontend development, contributing to various projects that included landing pages, e-commerce platforms, and content management systems. Throughout this period, proficiency was strengthened in React.js, Next.js, Tailwind CSS, and Shadcn UI, alongside practical experience with complementary tools such as React Hook Form, Zod, TanStack Table, and TanStack Query.

Collaboration with fellow interns and integration with backend APIs deepened the understanding of team-based workflows, version control using Git, and the importance of effective communication in software development. The experience further enhanced problem-solving capabilities, attention to detail, and the ability to deliver responsive, interactive, and dynamic web applications.

Overall, the internship substantially advanced technical expertise, provided meaningful professional insight, and fostered increased confidence in approaching future challenges within web development.

4.2 Learning Outcomes

The internship at Global Square IT Company facilitated the acquisition of a wide array of technical and professional competencies, significantly deepening the understanding of frontend development and enhancing readiness for real-world software engineering challenges.

- **Frontend Development Skills:** Gained hands-on experience in building responsive, interactive, and visually appealing web applications using React.js, Next.js, Tailwind CSS, Shadcn UI, and vanilla CSS.
- **Form Handling and Validation:** Learned to implement robust form handling and validation using React Hook Form and Zod, ensuring accurate and user-friendly input management.
- **Data Management and Dynamic UI:** Gained experience in integrating frontend components with backend APIs, fetching dynamic data, and displaying it effectively using tools like TanStack Table and TanStack Query.

- **Version Control and Collaboration:** Developed practical skills in Git and GitHub, including branching, merging, resolving conflicts, and collaborating efficiently with a team of interns and developers.
- **Problem Solving and Debugging:** Improved abilities to identify, troubleshoot, and fix UI/UX bugs, ensuring consistency, performance, and responsiveness across devices.
- **Project Management and Workflow:** Learned to work in a structured project environment, manage tasks, prioritize features, and follow best practices for code organization and maintainability.

REFERENCES

- JavaScript.info. (2024). The modern JavaScript tutorial. <https://javascript.info>
- shadcn/ui. (2024). Shadcn/ui: Beautiful, customizable React components. <https://ui.shadcn.com>
- TanStack. (2024a). TanStack Query: Powerful asynchronous state management. <https://tanstack.com/query/latest>
- TanStack. (2024b). TanStack Router: Type-safe routing for modern web applications. <https://tanstack.com/router/latest>
- TanStack. (2024c). TanStack Table: Headless UI for building powerful tables and datagrids. <https://tanstack.com/table/latest>
- W3Schools. (2024). JavaScript tutorial. <https://www.w3schools.com/js/>
- Zod. (2024). Zod documentation. <https://zod.dev>

APPENDIX

Hoslog Affiliate Landing Page



ホスログのアフィリエイト
プログラムとは

ホストクラブの初回予約に特化したプラットフォーム「ホスログ」のアフィリエイトプログラムは、専用の紹介コードを広告することでホスログの新規利用者を獲得し、ホストクラブへの予約が実行されるとその件数により紹介報酬が獲得できるプログラムです。

1件のネット予約成立につき
¥1,000報酬が獲得できます

新規アフィリエイターの獲得で
その収益の10%も受け取れる！

ホストクラブは審査済みの優良店！
安心・安全な広告活動ができます

アフィリエイター登録はこちら

Tips: そもそもアフィリエイトって？

Webサイト・ブログ・SNSなどのメディア運営者が、自分のメディアに広告を掲載し、そのメディアの閲覧者が広告から商品を購入したりやサービスを利用することで報酬が発生する仕組みです。

自身のメディアに広告を掲載
Webサイト・ブログ・SNSなどのメディアに商品やサービスの広告を掲載。

知人やSNSで商品を紹介
メールやLINEを使って、友人・知人に直接紹介URLやQRコードを共有。

YouTubeなどの動画で紹介
YouTubeやTikTokなど、動画を利用して視覚的に広いターゲットにプローチする近年主流の手法です。

ホスログアフィリエイトシステムは
こんな方におすすめ！

-  ホストクラブを普段から利用している方
(飲食店・美容室・バーなど)
-  SNSインフルエンサー
-  Webサイトやブログなどメディアを運用している方
-  アフィリエイトで新しい案件をお探しの方
-  ホストクラブ周辺に店舗を構えるオーナー様
(飲食店・美容室・コンビニなど)

ホスログ独自報酬の仕組み

Base System

紹介した利用者がホストクラブへ予約・来店すると
1予約につき1,000円の成果報酬が発生！

「報酬パターン 01」



専用コードで広告活動

あなたの予約獲得件数

予約数 × ¥1000 獲得！

1 「ホスログ」を専用コードで紹介

2 予約が1件入ったので¥1,000GET

3 追加で3件予約!! +¥3,000

4 紹介したユーザーの予約報酬は何度も貰えるんだね!!

さらに、継続報酬プログラムを採用！

成果対象のユーザーが更に他の人を紹介した際に自分にも報酬が!!

「報酬パターン 02」

実際に活用した収益イメージ

様々な方やシーンで紹介を行った場合の紹介人数・収益イメージです

 <p>報酬パターン① 事務職員Aさんの例 友人や自身のブログ経由で紹介</p> <table border="1" style="margin: auto;"> <thead> <tr> <th>予約件数</th> <th>10件 (¥1,000×10)</th> </tr> </thead> <tbody> <tr> <td>メンバーAの予約獲得件数</td> <td>10件</td> </tr> <tr> <td>メンバーBの予約獲得件数</td> <td>0件</td> </tr> </tbody> </table> <p>¥10,000/月</p>	予約件数	10件 (¥1,000×10)	メンバーAの予約獲得件数	10件	メンバーBの予約獲得件数	0件	 <p>報酬パターン② インフルエンサーBさんの例 自身の店舗や知り合い経由で紹介</p> <table border="1" style="margin: auto;"> <thead> <tr> <th>予約件数</th> <th>40件 (¥1,000×40)</th> </tr> </thead> <tbody> <tr> <td>メンバーAの予約獲得件数</td> <td>240名 (¥100×240)</td> </tr> <tr> <td>メンバーBの予約獲得件数</td> <td>0件</td> </tr> </tbody> </table> <p>¥64,000/月</p>	予約件数	40件 (¥1,000×40)	メンバーAの予約獲得件数	240名 (¥100×240)	メンバーBの予約獲得件数	0件	 <p>報酬パターン③ 飲食店店長Cさんの例 自身の各種SNSや知人経由で紹介</p> <table border="1" style="margin: auto;"> <thead> <tr> <th>予約件数</th> <th>78件 (¥1,000×78)</th> </tr> </thead> <tbody> <tr> <td>メンバーAの予約獲得件数</td> <td>624名 (¥100×624)</td> </tr> <tr> <td>メンバーBの予約獲得件数</td> <td>240名 (¥100×240)</td> </tr> </tbody> </table> <p>¥164,400/月</p>	予約件数	78件 (¥1,000×78)	メンバーAの予約獲得件数	624名 (¥100×624)	メンバーBの予約獲得件数	240名 (¥100×240)
予約件数	10件 (¥1,000×10)																			
メンバーAの予約獲得件数	10件																			
メンバーBの予約獲得件数	0件																			
予約件数	40件 (¥1,000×40)																			
メンバーAの予約獲得件数	240名 (¥100×240)																			
メンバーBの予約獲得件数	0件																			
予約件数	78件 (¥1,000×78)																			
メンバーAの予約獲得件数	624名 (¥100×624)																			
メンバーBの予約獲得件数	240名 (¥100×240)																			

選ばれる3つのポイント

Point 01

自身のSNSで広範囲に拡散可能

自身のSNSで「ホスログ」や「ホスログアフィリエイトプログラム」を紹介することで友人・知人を始め広範囲に拡散!
あなた専用の「利用者向け」「アフィリエイター向け」2種類の紹介コードを使ってSNS発信してください。



Point 02

リアルな店舗も宣伝ができる

ホストクラブ・近辺の居酒屋・美容室・コンビニなど、オンラインだけでなくリアルな店舗でもアフィリエイトプログラムによる広告活動が可能です!設置場所毎に紹介コードをカスタマイズでき複数店舗を所有のオーナー様にお薦め!



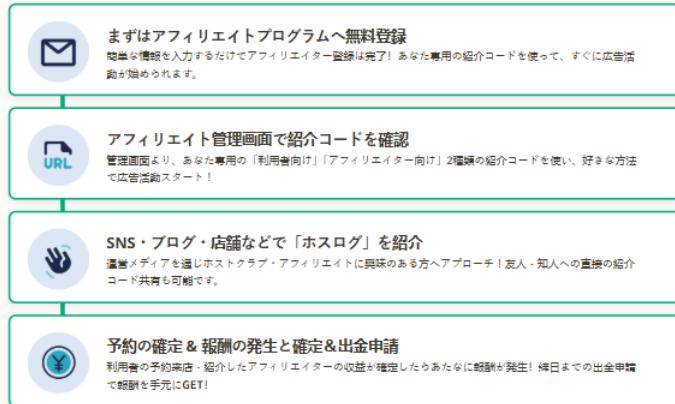
Point 03

待っているだけで継続収入が入る

自身で広告活動を行わずとも、あなたが紹介したアフィリエイター、更にアフィリエイターが紹介したアフィリエイターが広告活動により獲得した報酬の10%は待っているだけで継続的にあなたにも加算されます。(第二階層まで)



報酬獲得までの流れ



一部機能のご紹介

登録・利用は完全無料！
振込手数料を差し引いて、
成果報酬は全額入金されます

利用料金 0 円

ダッシュボード データ分析



\ 登録無料で簡単に広告活動が始められる！ /

hoselog.jp
ホスログのアフィリエイトプログラムに今すぐご参加ください！

無料登録をして今すぐ広告活動を始める

アフィリエイター利用規約 | プライバシーポリシー

© 2019 - 2026 ホスログ

Tapaiko Bazar

The screenshot shows a web browser window with the URL `localhost:5173/login` in the address bar. The page title is "Tapaiko Bazar". The main content is a "Login" form. It includes fields for "Email" (placeholder: "Enter your email") and "Password" (placeholder: "Enter your password"). Below the password field is a link "Forgot Password?". A green "Login" button is centered at the bottom of the form.

Login

Email
Enter your email

Password
Enter your password

[Forgot Password?](#)

Login

The screenshot shows a modal dialog box titled "Add Vendor". The dialog contains fields for "Company Name" (placeholder: "Enter company name"), "Contact Person" (placeholder: "Enter contact person"), "Email" (placeholder: "Enter email"), "Phone" (placeholder: "Enter phone number"), and "Address" (placeholder: "Enter address"). At the bottom right of the dialog is a green "Add Vendor" button.

Company Name
Enter company name

Contact Person
Enter contact person

Email
Enter email

Phone
Enter phone number

Address
Enter address

Add Vendor

Vendors

Add Vendor

Vendors Table

All Vendors Active Vendors Inactive Vendors

Vendor Id	Company Name	Contact Person	Email	Phone	Address	Joined On	Status	Actions
1	Samsung Nepal	Utsarga	utsarga@gmail.com	9801122334	Kalimati	2026-01-09	● Active	
2	Local Books Supplier	Amar	limbuamardeep18@gmail.co	9821313412	Kapan	2026-01-09	● Inactive	
3	Bienawf	Claref	contact@claref.com	9801122334	string	2026-01-09	● Active	
4	Brand Nepal	Bikin Nepal	bikinnepal@gmail.com	9843522535	Kapan	2026-01-09	● Active	

< Previous 1 Next >

Product Management

Add New Product

Basic Details

Product Name

Short Description

Description

Highlight this product in featured section

Upload Product Details

Upload Image

Categories

Product Categories

Attributes

Color Size RAM Weight
 Storage Type Capacity Shoes size
 Lens Quantity

Publish Product

Men's Summer Bomber Windproof Jacket.



Stay stylish and protected with this Men's Summer Bomber Windproof Jacket, designed for warm-weather comfort and reliable wind resistance. Crafted from lightweight, breathable fabric, this jacket helps block wind while keeping you cool and comfortable throughout the day. The classic bomber-style design adds a modern, sporty edge, making it easy to pair with jeans, chinos, or joggers. Ribbed cuffs and hem ensure a snug fit, while the smooth zipper closure allows for easy wear. Perfect for casual outings, travel, and evening wear during summer and mild weather.

Rs. 899

Category: Clothes Stock: 10

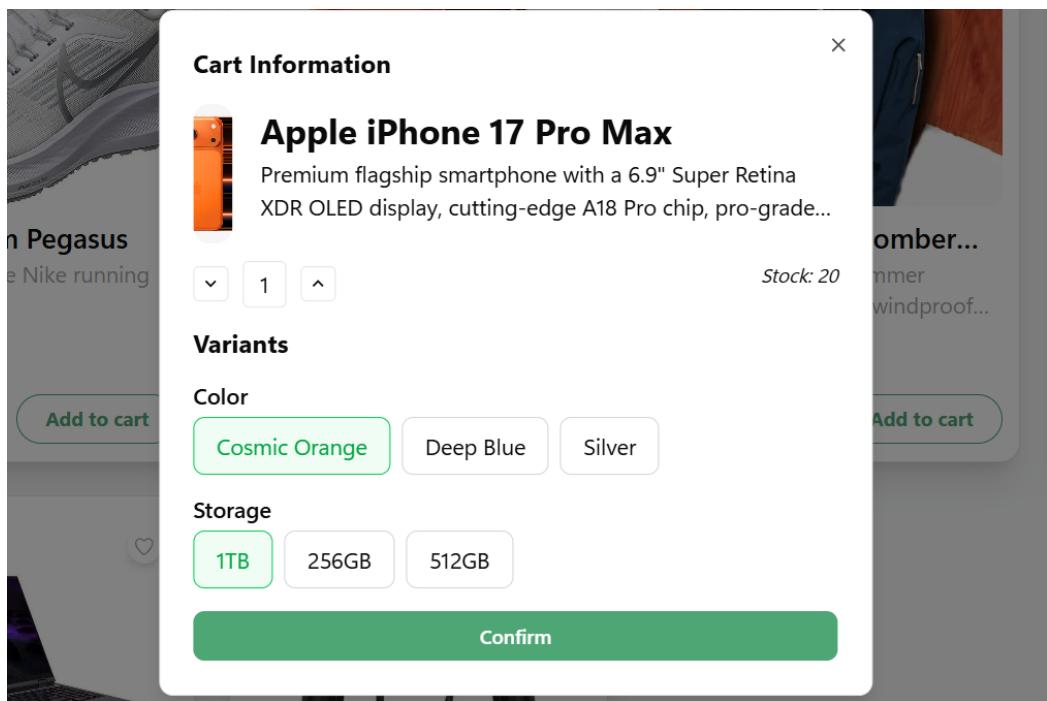
Variants

Size

Color

Add to cart

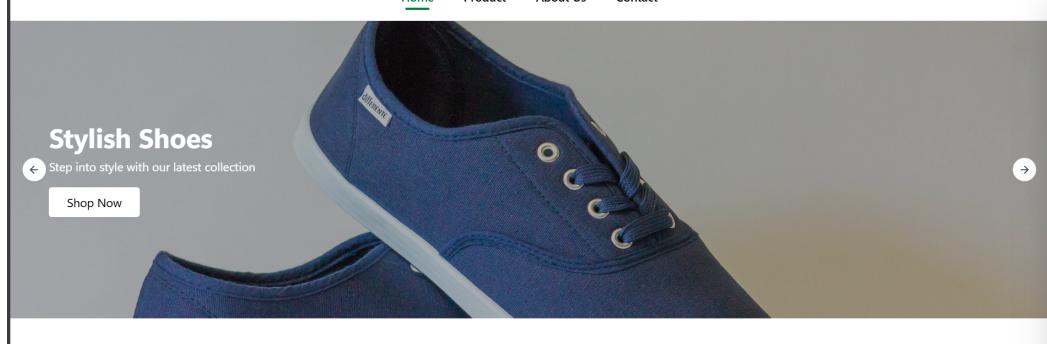
3 Issues



Our Happy Customers

Dont just take our word for it – see how our products and services have delighted customers across the globe, one experience at a time.

[Add Your Own Review](#)



localhost:3000/product?tags=4K+Camera%2CBest+Seller

The screenshot shows a web browser displaying a product search results page. The URL in the address bar is `localhost:3000/product?tags=4K+Camera%2CBest+Seller`. The page has a header with the logo "त्रिपुरा Bazaar", a search bar, and navigation links for Home, Product (which is underlined in green), About Us, and Contact. On the left, there's a sidebar with categories like All, Grocery, Clothes, Shoes, and Electronics, and a filter section for Tags (4K Camera, Best Seller, Electronics, Fashion) with a red badge showing '5 issues'. The main content area displays four products in cards: Canon Camera (Rs. 3000), LG Refrigerator (Rs. 89000), Reebok Nano X (Rs. 13000), and Puma Smash v2 (Rs. 10000). Each card includes a 'View Details' button and an 'Add to cart' button.

A modal dialog box titled "Forgot Password" is displayed. It contains a text input field labeled "Enter your email address" with the placeholder "your@email.com". Below the input field is a green "Submit Email" button. A close button ("X") is located in the top right corner of the modal.