Tribhuvan University

Faculty of Humanities and Social Sciences

A PROJECT PROPOSAL
ON
CLIENT MANAGEMENT SYSTEM

Submitted to

Department of Computer Application

Nepal Mega College

In partial fulfillment of the requirements for the Bachelors in Computer Application

Submitted by

Apeksha Dhungana

Reg no: 6-2-40-110-2021

Nov, 2025

ABSTRACT

The Client Management System (CMS) is a well-established concept aimed at efficiently managing the client lifecycle through advanced technologies and process-driven tools. This project focuses on streamlining client-related processes via a web-based application that enhances data organization, accessibility, and operational efficiency.

The CMS functions as both a centralized repository for client information and a tool for tracking interactions, managing workflows, and ensuring seamless client-business communication. This study emphasizes the development and implementation of a CMS, highlighting essential features such as client data management, interaction history tracking, and report generation to improve business productivity and client satisfaction.

By leveraging scalable and user-friendly web applications, the project showcases how web technologies can simplify client management processes while addressing the evolving needs of modern businesses. It further underscores the role of web-based solutions in fostering innovation and enhancing operational efficiency within competitive business environments.

Keywords: *Hashing Algorithm, Client Management System*

TABLE OF CONTENTS

LIST OF FIGURE

1.Introduction

A Client Management System (CMS) is a software application designed to manage a company's interactions with its current and potential clients. It serves as an essential tool for businesses to streamline client-related processes, ensuring efficient and organized handling of client data. By automating routine tasks, the CMS enhances productivity, reduces manual errors, and provides a structured approach to client lifecycle management.

The CMS developed in this project leverages **React**, **Node.js** as the server-side scripting language and **MySQL** as the database for data storage and management. This combination ensures a robust, scalable, and high-performing application capable of delivering rapid data processing in a systematic manner. The system maintains detailed records of clients, enabling businesses to access and update client information efficiently while ensuring data consistency and security.

The project is designed with two distinct modules:

1. **Admin Module**: This module allows administrators to manage client data, oversee system functionalities, and generate reports. Administrators have complete control over the CMS to ensure smooth operations.

2. **Client Module**: This module provides clients with limited access, allowing them to view or interact with their specific data as required.

2.Problem Statement

Many businesses face inefficiencies because client data is managed manually or across multiple platforms. This leads to inconsistent information, errors, and duplicated records. Employees spend excessive time searching for client details and tracking interactions, reducing productivity. Communication with clients often becomes unclear or delayed due to the lack of a centralized system. Additionally, generating accurate reports is time-consuming and unreliable. These challenges prevent businesses from making timely decisions and delivering effective, consistent client service.

3.Objectives

The main objectives of this project are mentioned below:

- To automate client management for efficient storage and retrieval of client details.

- To save time, energy, and money by eliminating manual client data searches.
- To streamline services and client information for better organization and management.

4.Methodology

The development of the Client Management System (CMS) follows a structured approach using the Waterfall Model. [1] This model ensures a systematic and sequential progression through the development phases, providing a clear framework for planning and execution. The Waterfall Model consists of the following stages:

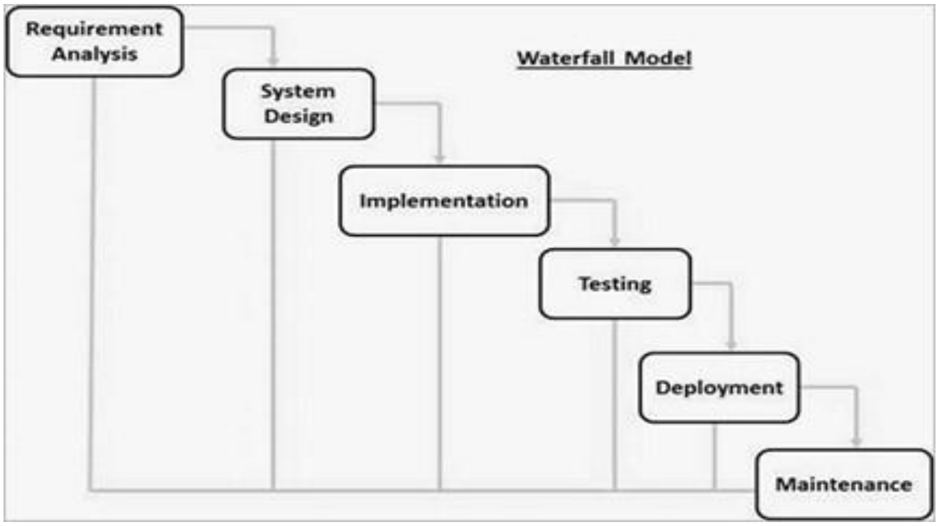


Figure 1:Waterfall Model of CMS

- **Requirement Analysis** – Collecting user and business requirements to define the essential features of the Client Management System. Feedback from stakeholders ensures that all needs are clearly understood, realistic, and aligned with business goals.
- **System Design** – Creating the system architecture, database structure, and user interface layout. Continuous feedback helps refine workflows, ensuring the CMS is user-friendly and efficient.
- **Implementation** – Developing the CMS frontend using React.js and the backend using Node.js. Features are built and improved iteratively based on ongoing feedback.

- **Testing** – Performing unit, integration, and system testing to identify and fix errors. Feedback from testers and users ensures the system functions smoothly and meets requirements.
- **Deployment** – Releasing the CMS for real-world use. User and stakeholder feedback helps fine-tune the system for better performance and usability.
- **Maintenance** – Continuously monitoring and updating the system to improve security, performance, and usability.

a. Requirement Identification

The development of a **Client Management System (CMS)** for businesses involves identifying and defining the key requirements and functionalities necessary to streamline client interactions, improve data management, and enhance operational efficiency.

i. Study of Existing System

Many businesses currently manage client data through a variety of manual processes or disparate systems, leading to inefficiencies and errors. Existing systems, such as CRM software, offer some automation but often lack the full integration and scalability needed for effective client management. These systems typically feature basic functionalities like client contact management, interaction history tracking, and task management. [2] However, they often fail to provide a comprehensive, centralized platform that integrates with other business processes, such as reporting, analytics, and communication tools. As a result, businesses face challenges such as inconsistent data, missed opportunities, and inefficient workflows.

The proposed Client Management System (CMS) aims to improve upon these existing solutions by creating a fully integrated, web-based platform that centralizes all client data, automates key workflows, and enhances communication channels between businesses and their clients.

ii. Literature Review

A study by McDonald and McDonald [J. McDonald and R. McDonald, "Client Relationship Management: A Comprehensive Approach," Business Journal, vol. 25, no. 4, pp. 235-248, Aug. 2003.] highlights the importance of integrating client data with other business functions to enhance productivity. The authors emphasize the need for systems that allow businesses to track interactions in real-time, helping improve response times and customer satisfaction. Another study by Smith and Jones [A. Smith and B. Jones, "Advances in Client Relationship Management Systems," Journal of Business Technology, vol. 33, no. 2, pp. 112-118, Jan. 2007.] focuses on the benefits of automated reporting within CMS platforms. By automating data entry and report generation, businesses can make better, data-driven decisions, improving overall efficiency.

iii. Requirement Analysis

The requirement analysis for the **Client Management System (CMS)** focuses on understanding the needs of both users and the system to ensure effective implementation. This process involves defining the key functionalities, user experience expectations, and technical specifications that will guide the development of the CMS.

User Needs

- Business Owners/Managers: A centralized platform to manage all client information, track interactions, and generate reports.
- Sales and Marketing Teams: Access to detailed client data for targeted campaigns, tracking communication history, and managing follow-ups.
- Customer Support Teams: A system that allows quick access to client issues and interactions for efficient case management and support.
- Clients: Efficient communication with the business, easy access to service updates, and personalized offers based on interaction history.

System Functionalities

- Client Data Management: Store and organize detailed client profiles, including contact details, purchase history, and preferences.
- Interaction History Tracking: Track all communication with clients, ensuring businesses can review past interactions to improve service.
- Report Generation: Automatically generate reports on client data, interactions, and business performance.
- User Notifications: Alert users about important updates, such as upcoming client meetings, follow-ups, or new client inquiries.

- Scalable Architecture: Ensure the system can handle growing amounts of client data and adapt to the evolving needs of the business.
- Security Features: Implement strong security measures to protect sensitive client data.

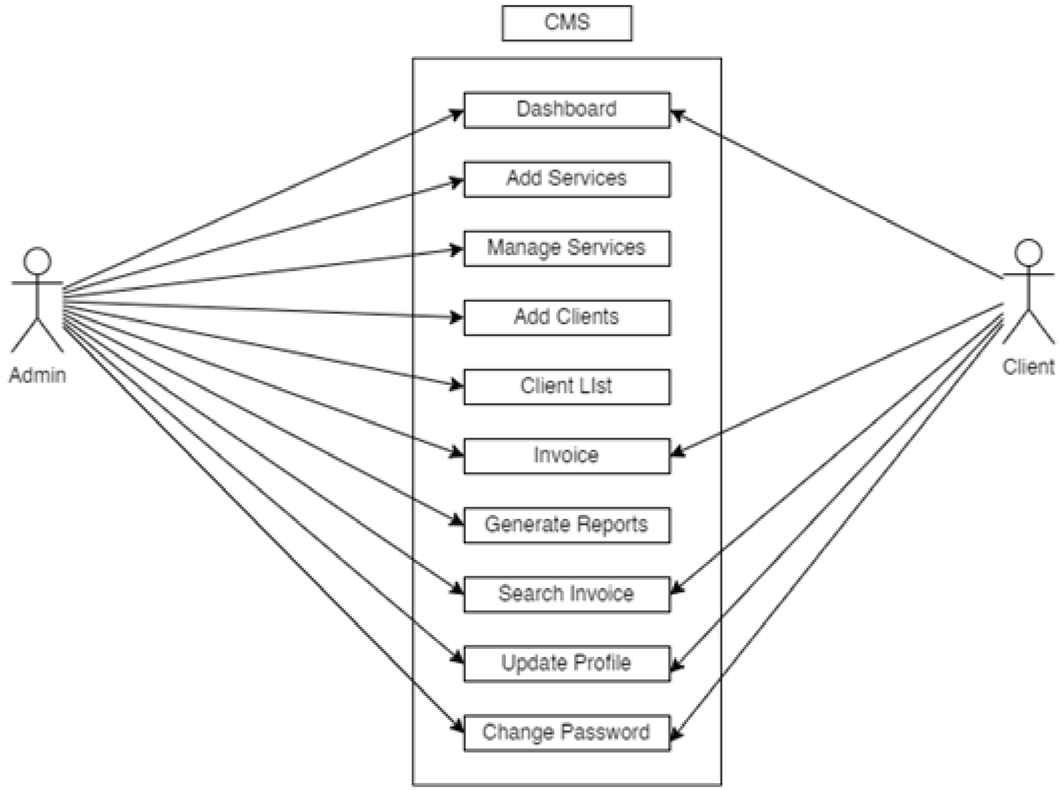


Figure 2: Use Case diagram of CMS

The use case diagram shows the core functionalities of the CMS for both admin and client roles. Admins can manage services, clients, invoices, reports, and their own profile. Clients can view services, access invoices, and update their profile. This clear role separation ensures efficient system management, secure access, and smooth interactions within the Client Management System.

b. Feasibility Study

The feasibility study for the Client Management System (CMS) evaluates the project's viability from technical, operational, economic, and scheduling perspectives.

i. Technical Feasibility
Evaluates the necessary technology stack for data integration and system scalability.

ii. Economic Feasibility
Analyzes development costs versus the benefits, like reduced travel time and congestion.

iii. Operational Feasibility
Assesses the ease of use and integration with existing systems.

c. High Level Design of the System

i. Methodology of the proposed System

This is the step-by-step approach for building the CMS. It includes understanding requirements, designing the system, developing it, testing for errors, deploying it for users, and maintaining it over time.

ii. Flow Chart

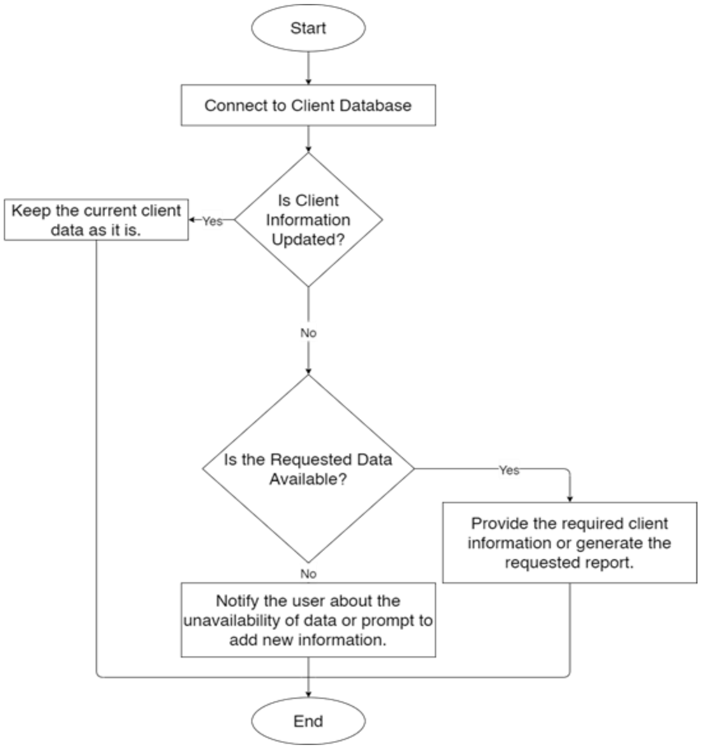

Start
Connect to Client Database
Is Client Information Updated?
Yes
Keep the current client data as it is.
No
Is the Requested Data Available?
Yes
Provide the required client information or generate the requested report.
No
Notify the user about the unavailability of data or prompt to add new information.
End

Figure 3: Flow Chart of CMS

iii. Working Mechanism and Tools

This section explains how the Client Management System (CMS) operates. The system focuses on collecting client data, organizing and storing it securely, displaying detailed client information to authorized users, automating workflows such as report generation, and updating client records based on interactions and user inputs.

- **React.js**

React is used to build the interactive user interface of the CMS. It enables a fast, responsive, and component-based front-end where users can view, search, update, and manage client information efficiently.

- **Node.js**

Node.js powers the backend of the CMS. It handles server-side logic, processes client requests, manages authentication, performs form validations, and interacts with the database to securely store and retrieve client data.

- **MySQL**

MySQL is used to store all client records, communication logs, reports, and system-generated data. It ensures secure, scalable, and structured data management with features like indexing and relational table connections.

- **Draw.io**

Draw.io is used to design system diagrams such as use case diagrams, DFDs, flowcharts, and ER diagrams. These diagrams help visualize how various components of the CMS interact and function.

- **MS Word**

MS Word is used for preparing documents including project reports, system design documentation, and user manuals. It supports clear formatting for presenting project details.

iv. ER-Diagram

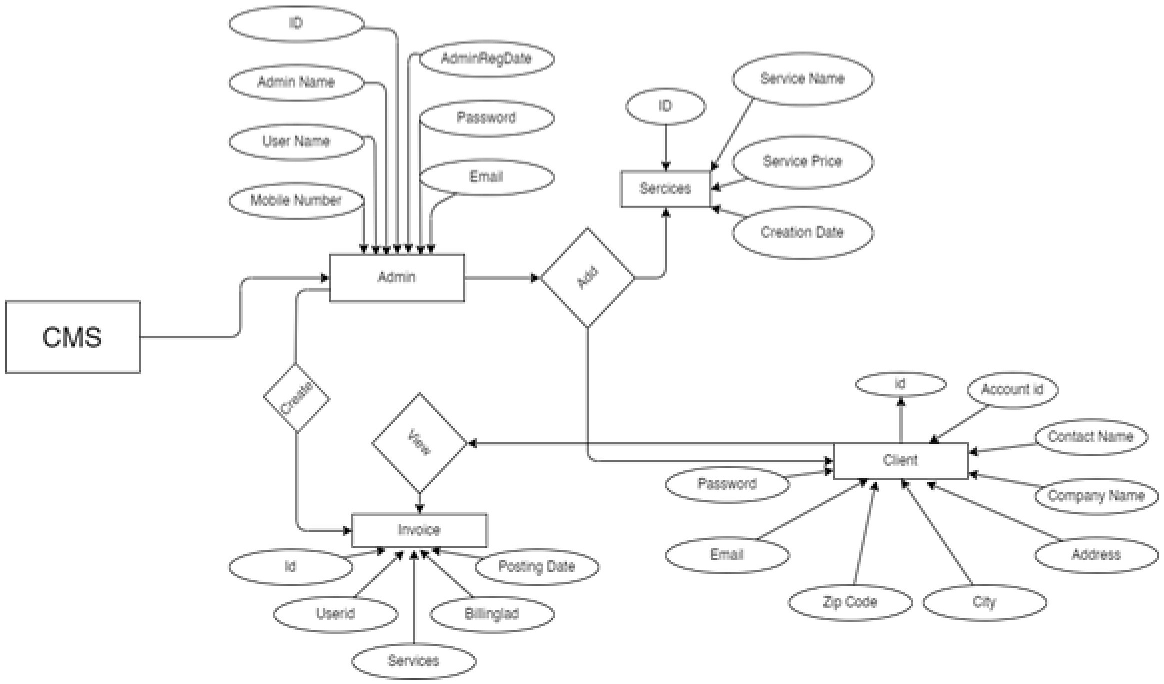


Figure 4: ER-Diagram of CMS

The image is an entity-relationship (ER) diagram representing a simplified data model for a dating site. It consists of three entities: **User**, **Address**, and **Interest**. The **User** entity includes attributes such as **Name, Age, Gender, Phone Number, E-mail Address, and Photo**. The **Address** entity contains **City, Country, and Validate**, with an arrow indicating a relationship where a user is associated with an address. The **Interest** entity has a single attribute, **User Interest**, and is linked to the **User** entity, indicating that each user can have associated interests. This model provides a basic structure for storing user profiles, location data, and preferences within the dating platform.

v. Process Modeling

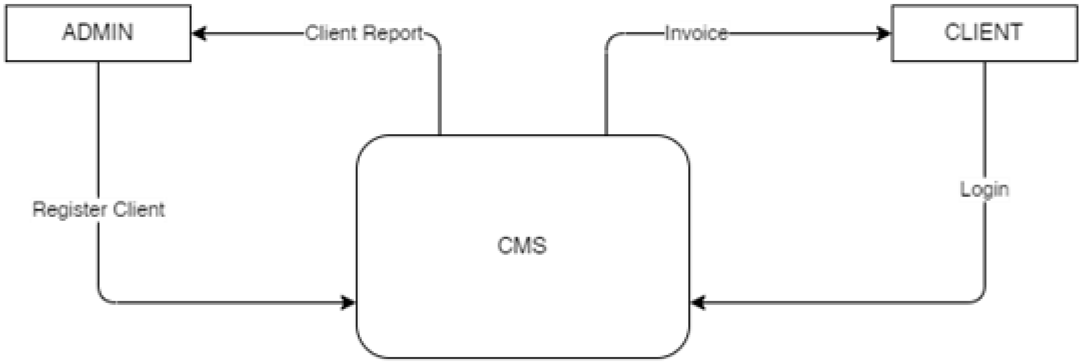


Figure 5: DFD-1 of CMS

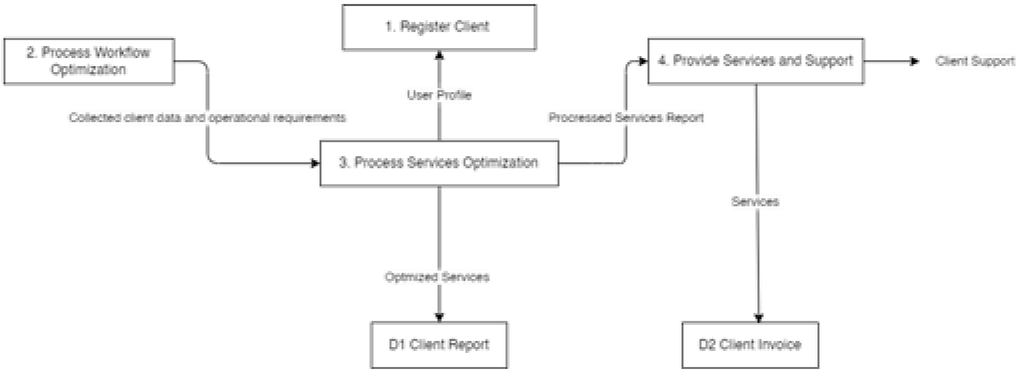

1. Register Client
2. Process Workflow Optimization
Collected client data and operational requirements
User Profile
3. Process Services Optimization
Processed Services Report
4. Provide Services and Support
Client Support
Services
Optimized Services
D1 Client Report
D2 Client Invoice

Figure 6: DFD-2 of CMS

vi. Description of Algorithm

1. Hashing Algorithm

In a Client Management System, a hashing algorithm is used to secure sensitive user data, mainly passwords. When an admin or client creates an account or logs in, the system converts their password into an irreversible hashed value using algorithms such as bcrypt, SHA-256, or Argon2. Instead of storing the actual password, only the hashed output is saved in the database. During login, the system hashes the entered password again and compares it with the stored hash to verify the user. This process ensures that even if the database is compromised, the original passwords cannot be retrieved, enhancing overall system security and protecting user information from unauthorized access. [3]

Key Features of the Dating Platform

1. **Secure Password Storage**
   Hashing ensures that actual passwords are never stored in the database. Instead, only the hashed versions are saved, protecting user credentials even if the database is accessed by unauthorized users.
2. **Safe Authentication Process**
   During login, the system hashes the entered password and compares it with the stored hash. This provides secure verification without revealing or exposing the actual password.
3. **Protection Against Data Breaches**

Hashing algorithms like bcrypt and Argon2 make it extremely difficult for attackers to reverse the hash. Even in case of a data leak, passwords remain unreadable, enhancing system security and protecting client information.

How Hashing Algorithm Works in a Client Management System

1. User Password Conversion into Hash

- When a client or admin creates an account or updates their password, the CMS converts the plain-text password into a **hashed value** using a hashing algorithm like **bcrypt, SHA-256, or Argon2**. This transforms the password into a fixed-length, irreversible string, ensuring it cannot be easily read or recovered.

2. Storing Hashed Passwords

- CMS stores **only the hashed password** in the database, never the plain-text version.
- This protects user credentials in case of a data breach.

3.Authentication During Login

- When the user logs in, the system hashes the entered password using the same algorithm.
- The new hash is compared with the stored hash.
- If the hashes match, the user is authenticated; if not, access is denied.

4.Security Benefits

- Passwords remain secure even if the database is compromised.
- Hashing ensures **one-way encryption**, meaning original passwords cannot be reverse-engineered.
- Protects the CMS from unauthorized access and builds user trust.

5.Gantt Chart

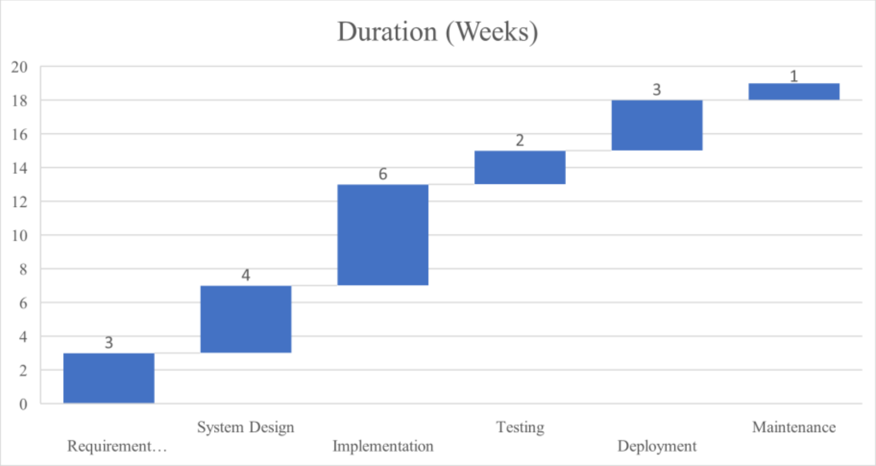


Figure 7: Gantt Chart

6.Expected Outcome

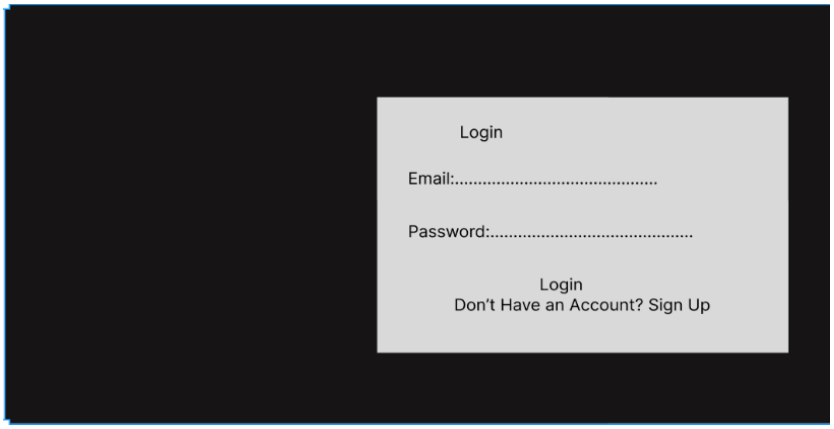
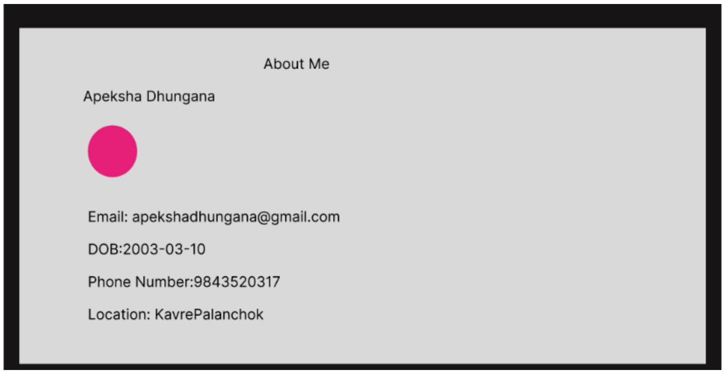

Login
Email:..
Password:..
Login
Don't Have an Account? Sign Up

Figure 8: Login

Admin Profile

Figure 9: Admin Profile



About Me

Apeksha Dhungana

Email: apekshadhungana@gmail.com

DOB:2003-03-10

Phone Number:9843520317

Location: KavrePalanchok

Figure 10: User profile

References

[1] R. K. Z. S. P. Y. D. M. A. Sabahat Ansari, "Client Management System," *International Journal of Adva,* 2023.

[2] A. Z. H. T. R. O. D. S. H. M. Samppa Suoniemi, The journal of business and cms, 2022.

[3] D. N. R. M. Gokul, Hashing Algorithm for Load Balancing in Cloud Server, 2025 6th International Conference on Mobile Computing and Sustainable Informatics (ICMCSI).