**Tribhuvan University**

## Faculty of Humanities and Social Sciences

**A PROJECT REPORT ON**

**"Sayogi"**

**Submitted to**

**Department of Computer Application**

**Aadim National College**

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

**Submitted by**

Binod Gautam.

2080/02/26

Under the Supervision of

**Mr.Sanjay Shah**

**Tribhuvan University**

**Faculty of Humanities and Social Sciences**

**Aadim National College**

## Supervisor's Recommendation

I here by recommend that this project prepared under my supervision by **Binod Gautam.** entitled "**Sayogi"** in partial fulfillment of the requirements for the degree of Bachelor of Computer Application is recommended for the final evaluation.

……………………

**SIGNATURE**

**Mr. Sanjay Shah**

**SUPERVISOR**

Chabahil, Chuchepati, Kathmandu 44600

**Tribhuvan University**

**Faculty of Humanities and Social Sciences**

**Aadim National College**

## LETTER OF APPROVAL

This is to certify that this project prepared by **Binod Gautam.** Entitled "**Sayogi**" in partial fulfillment of the requirements for the degree of Bachelor's in Computer Application has been evaluated. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

………………………..                                            …………………………..
**Mr. Sanjay Shah**                                          **Mr. Yuba Raj Kalathoki**
**Supervisor**                                                   **Coordinator**
Chabahil, Chuchepati,                                      Chabahil, Chuchepati
Kathmandu 44600                                            Kathmandu 44600

……………………….                                            ……………………….
**Internal Examiner**                                       **External Examiner**

# Abstract

**Sayogi** is an innovative web-based platform designed to seamlessly connect **service seekers** and **providers** within a structured waterfall development model. With a strong emphasis on **user-centric design**, the project aims to simplify the user experience by leveraging the **MERN stack, Tailwind CSS,** and **Mongoose.** The methodology involves rigorous testing, encompassing both **unit and system testing**, ensuring the reliability and responsiveness of the platform. The project's **objective** is to create a user-friendly environment that facilitates effective collaboration. Through strategic technology selection, **open communication**, and a commitment to continuous improvement, Sayogi has successfully achieved its goals. Future recommendations include prioritizing **user feedback**, enhancing analytics, and proactive scalability planning. In conclusion, **Sayogi** stands as a testament to the power of user-centric design and agile practices within a waterfall development model, ensuring a robust, adaptable, and user-friendly platform for seamless service connections.

# Acknowledgement

We sincerely acknowledge and thank to those who have contributed their valuable time in helping us to achieve the success in our project work. We would like to express our gratitude to all those gave us the possibility to complete this project. We want to thank Aadim National College for assigning **Yuba Raj Kalathoki** as a project head and also for his valuable and timely guidance, co-operation, encouragement & time spent for doing this project work.

We would also like to thank our Head of Department Shankar Prasad Sharma and project supervisor **Sanjay Shah** for their assistance and insightful comments, and who willingly shared their expertise with us. We are immensely obliged to our friends for their elevating inspiration, encouraging guidance and kind supervision in the completion of our project. We sincere thanks to the IT staff for providing us sufficient information which helped us to complete our project successfully. Last, but not the least, our parents are also an important inspiration for us. So, with due regards, we express our gratitude to them.

# Contents

# List of  Figures

# List of Tables

# Chapter 1: Introduction

## 1.1 Introduction

Sayogi is an innovative web-based application that provides a user-friendly platform for both service seekers and service providers. It enables service seekers to create detailed profiles that highlight their skills, experience, and expertise. Simultaneously, service providers can create service requests specifying their specific requirements.

Sayogi acts as a bridge between service seekers and providers, facilitating seamless connections and collaborations. By leveraging advanced search and filtering capabilities, users can easily find the right service request that matches their needs.

## 1.2 Problem statement

In today's service-oriented marketplace, there is a lack of a user-friendly and efficient platform that connects service seekers and service providers. Existing methods for finding service providers or service requests are often fragmented, time-consuming, and unreliable. This leads to difficulties for service seekers in identifying the right providers for their needs and for service providers in reaching their target audience. Consequently, there is a need for a platform like Sayogi that can address these challenges and provide a seamless experience for both service seekers and service providers.

Some of the major problems are:

- Difficulty in finding and Attracting Service Seekers.

- Inefficient Communication Channels.

## 1.3 Objectives

- To create a user-friendly platform that simplifies service discovery, builds trust, and streamlines service request management for a seamless experience between service seekers and providers.

## 1.4 Scope and Limitation

Scope refers to the range and extent of what a project or system aims to achieve. It outlines the boundaries and objectives, defining what is included within its parameters.

Limitations are constraints or restrictions that may impact the effectiveness or applicability of the project or system. They highlight factors that could hinder its performance or scope, setting boundaries on what it can achieve. Understanding both scope and limitations is crucial for a comprehensive assessment of any endeavor.

### Scope

- **Regional Reach**: Sayogi works better where more people use it, and it might not be as useful in less crowded areas.

- **Different Services for Success**: Sayogi does well when it has lots of different services. If it only focuses on a few, it might not be as helpful for people with different needs.

- **Easy Tech Access**: Sayogi needs the internet and mobile tech. If you don't have these, you might miss out, making it less inclusive and limiting its range.

### Limitation

- **Incomplete or Inaccurate User Profiles**: Users may not consistently provide detailed or accurate information, impacting the precision of matching service seekers with providers.

- **Limited User Engagement:** Varying levels of user participation in creating profiles or service requests can result in a reduced pool of available and accurate information on the platform.

- **Quality of Service Requests:** The effectiveness of connecting users depends on the specificity and quality of service requests. Poorly articulated requirements may lead to mismatches or difficulty in finding the right service provider.

## 1.5 Development Methodology

zMy proposed project will follow the Waterfall Model as its methodology for software development. This model presents a linear sequence of development phases, each dependent on the completion of the previous phase. Since I have a well-defined vision for the project, the Waterfall Model aligns with my goals and is the appropriate approach for its development
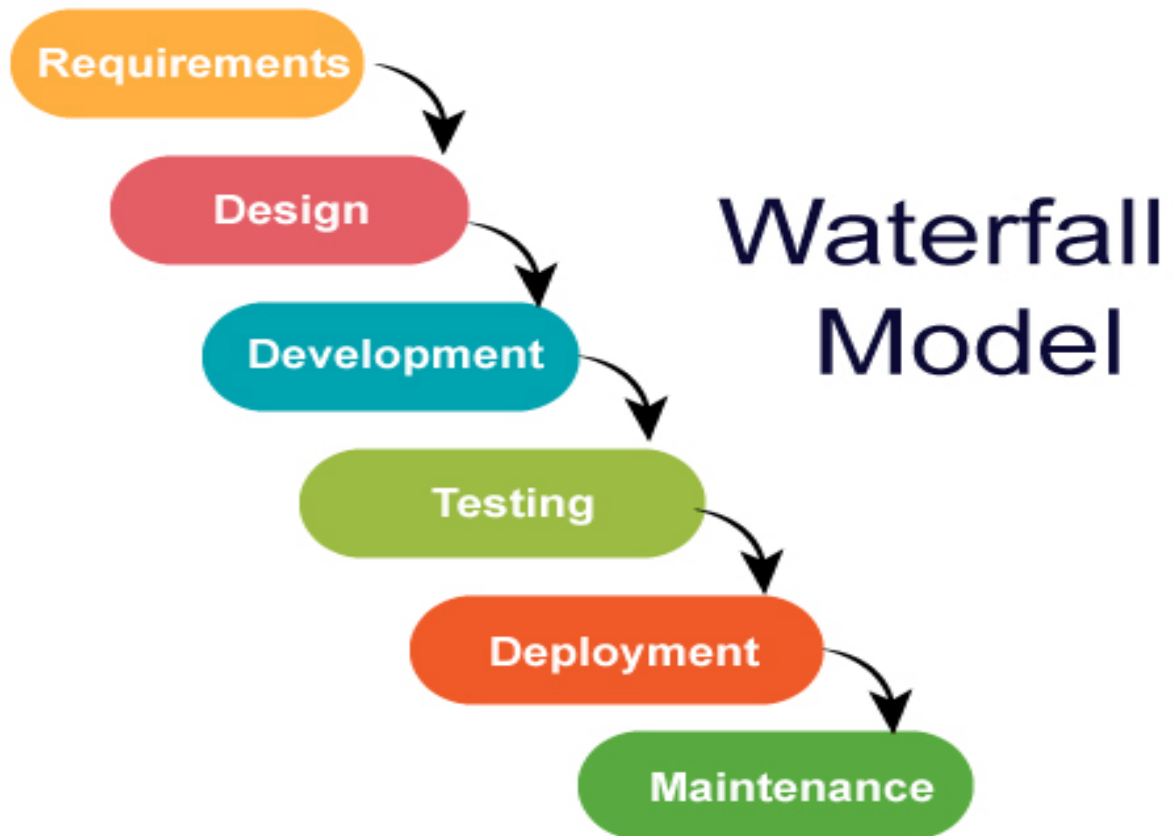


**Figure 1 : Waterfall Model**

## 1.6 Report Organization

This report document contains five chapters including this chapter. Chapter two defines and describes Background Study and Overview of the related existing systems and their pros and cons. Chapter three presents the System and Design including Requirement Analysis and Feasibility Study Chapter four presents the Implementation, Testing and Debugging. Conclusion, Limitations and future Enhancement are briefly explained.

# Chapter 2: Background Study and Literature Review

## 2.1 Background Study

The Background Study for Sayogi delves into fundamental theories and concepts within the service-oriented marketplace, focusing on the theoretical foundations of user-friendly web-based platforms connecting service seekers and providers. Key terminologies like "service discovery" and "trust-building" are explored, aiming to understand the essential features contributing to a seamless user experience.

This study also addresses existing challenges in the service-oriented marketplace, emphasizing issues like fragmentation and inefficient communication channels. The theoretical framework seeks to justify the need for a platform like Sayogi, positioned to overcome these challenges and provide a user-friendly and efficient experience for both service seekers and providers. Understanding these foundational elements sets the groundwork for the subsequent development of Sayogi, ensuring alignment with the core principles outlined in its introduction.

## 2.2 Literature Review

In Sayogi Literature Review, the examination of similar projects extends to innovative concepts. The platform considers personalized service requests, recommending opportunities based on users' profiles, enhancing efficiency and relevance. Furthermore, Sayogi explores the literature on establishing direct connections between service providers and seekers, aiming to simplify communication and collaboration by minimizing intermediaries.

Additionally, the review delves into the importance of user feedback by allowing service seekers to give ratings. This feature not only empowers users to share their experiences but also aids others in making informed decisions when selecting service providers. By amalgamating these elements, Sayogi aspires to integrate the best practices and inventive features, ensuring a personalized, direct, and user-friendly service experience for both seekers and providers.

# Chapter 3: System Analysis and Design

## 3.1 System Analysis

The System Analysis for Sayogi concentrates on refining the precision of connecting service seekers and providers by optimizing user profiles and skill-matching algorithms. The evaluation extends to assessing the efficiency and security of the direct connection feature, emphasizing the need for clear communication channels. Additionally, the analysis delves into improving the feedback loop through the implementation of a robust rating system.

This comprehensive assessment also includes a thorough examination of platform performance, scalability, and user experience, aiming to enhance navigation and visual design. Security measures are scrutinized to safeguard user data and ensure adherence to industry standards. The ultimate objective of this System Analysis is to identify strengths and areas for improvement, guiding the ongoing development and optimization of Sayogi towards an enhanced, secure, and user-friendly service experience.

## 3.1.1 Requirement Analysis

Requirement analysis is done while developing a system and before implementing it, it is necessary to analyze the whole system requirement. It is categories into mainly two parts:

- Functional requirements

- Non-functional requirements

The functional requirements are user "visible" features that are typically initiated by stakeholders of the system, such as generating reports, login, and sign up. On the other hand, non-functional requirements are requirements that describe how the system will do what it is supposed to do, for example, Usability, Reliability & Availability, Performance, Security and maintainability.

**i.Functional requirements**

- Admin

    - Login

    - View History

    - Manage User

- User

    - Login

    - Service Request

    - Create Service Seeeker account

    - Search Service Seeker

    - Add comments

    - Request to Hire

    - Get Notification

    - View History

- Visitors
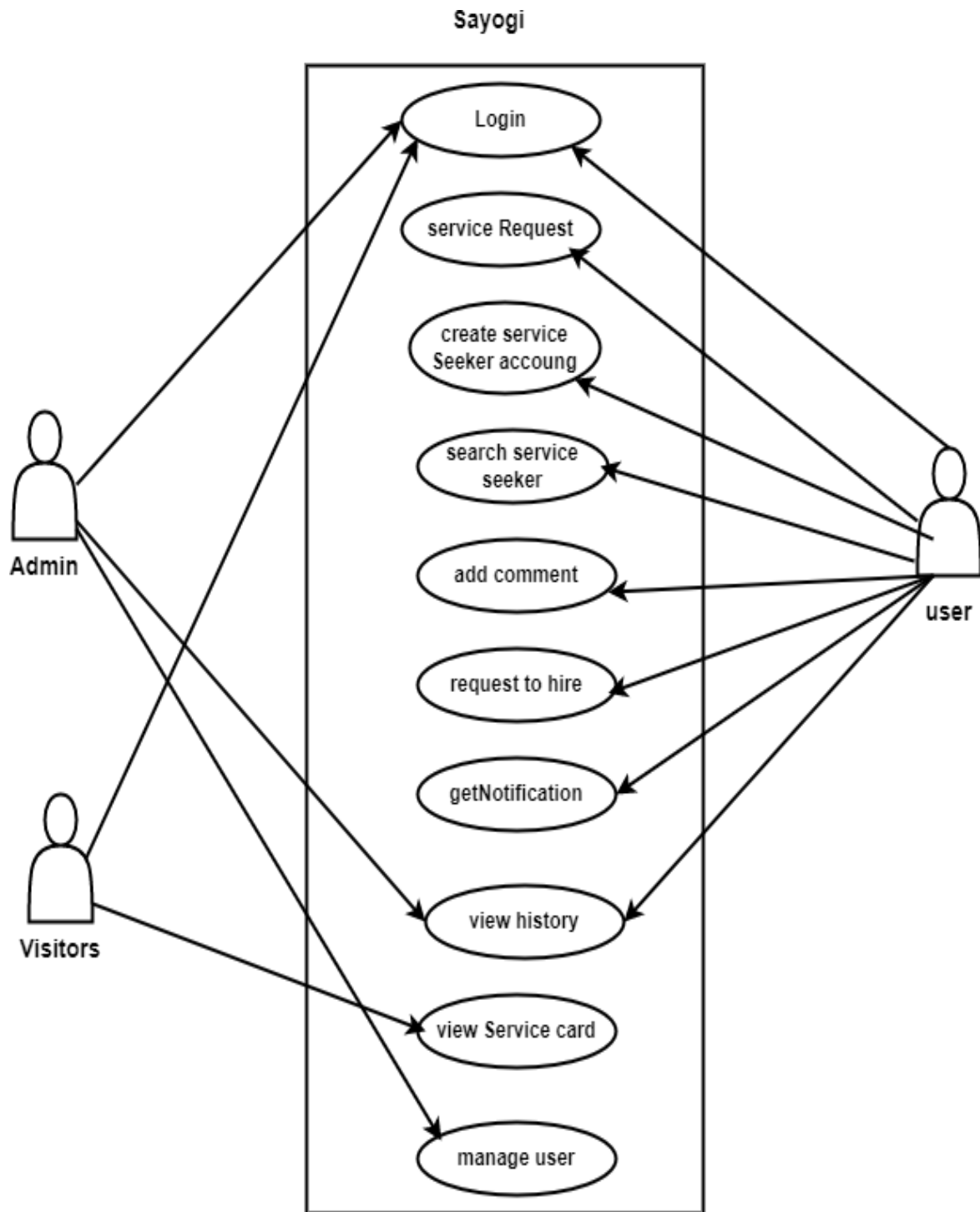
    - View Service Card

    - Login

**Figure 2: Use-Case diagram**

## ii. Non-functional Requirements

### 1. Security

- Implement proper authentication and authorization mechanisms to control user access
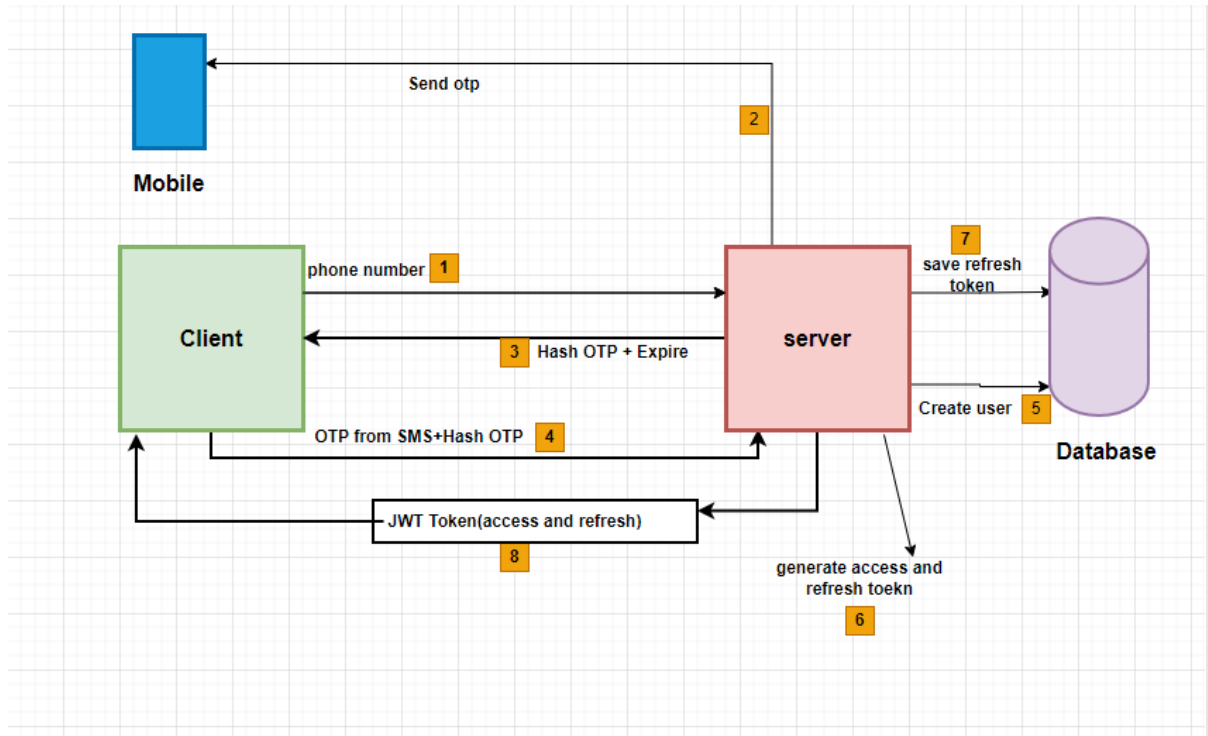


**Figure 3: Authentication**

### 2. Performance

- Sayogi should provide a responsive and performant user experience.

- Optimize system performance to handle concurrent user requests and minimize response times.

### 3. Scalability

- Sayogi should be designed to handle increasing user demands and accommodate future growth.

### 4. Compatibility

- Ensure Sayogi is compatible with various platforms, browsers, and devices.

**5. Maintainability**

- Write clean and well-documented code to enhance code maintainability.

## 3.1.2 Feasibility Study

After doing the project Sayogi, study and analysis all the existing or required functionalities of the system, the next task is to do the feasibility study for the project.

Feasibility study includes the consideration of all the possible ways to provide a solution the given problem. The proposed solution should satisfy all the user requirements and should be flexible enough so that future changes can be easily done based on the future upcoming requirements

**i. Technical feasibility**

The Technical Feasibility of the sayogi Nepal system has been evaluated, taking into account the necessary resources such as hardware, software, and human resources. It has been determined that all required resources are already available

**ii. Economic Feasibility**

The sayogi Nepal System utilizes open-source technologies, meaning there is no need for additional software or hardware. As a result, the only recurring cost associated with the system is the internet connection

**iii .Operational Feasibility**

The operational feasibility of the sayogi Nepal System was assessed, and we identified several challenges and vulnerabilities in the existing system. These issues were addressed during the development of the new system, resulting in a more robust and manageable platform for users. Detailed analysis and planning were conducted to ensure that the sayogi Nepal System is operationally feasible and meets the needs and expectations of its users.

## 3.1.3 Data Modeling (ER-Diagram)

This ER (Entity Relationship) Diagram represents the model of this project (sayogi). The entity-relationship diagram of the project shows all the visual instruments of the database table and the relations between admin, users, etc. It uses structured data to define the relationship between structured data groups of sayogi functionalities.
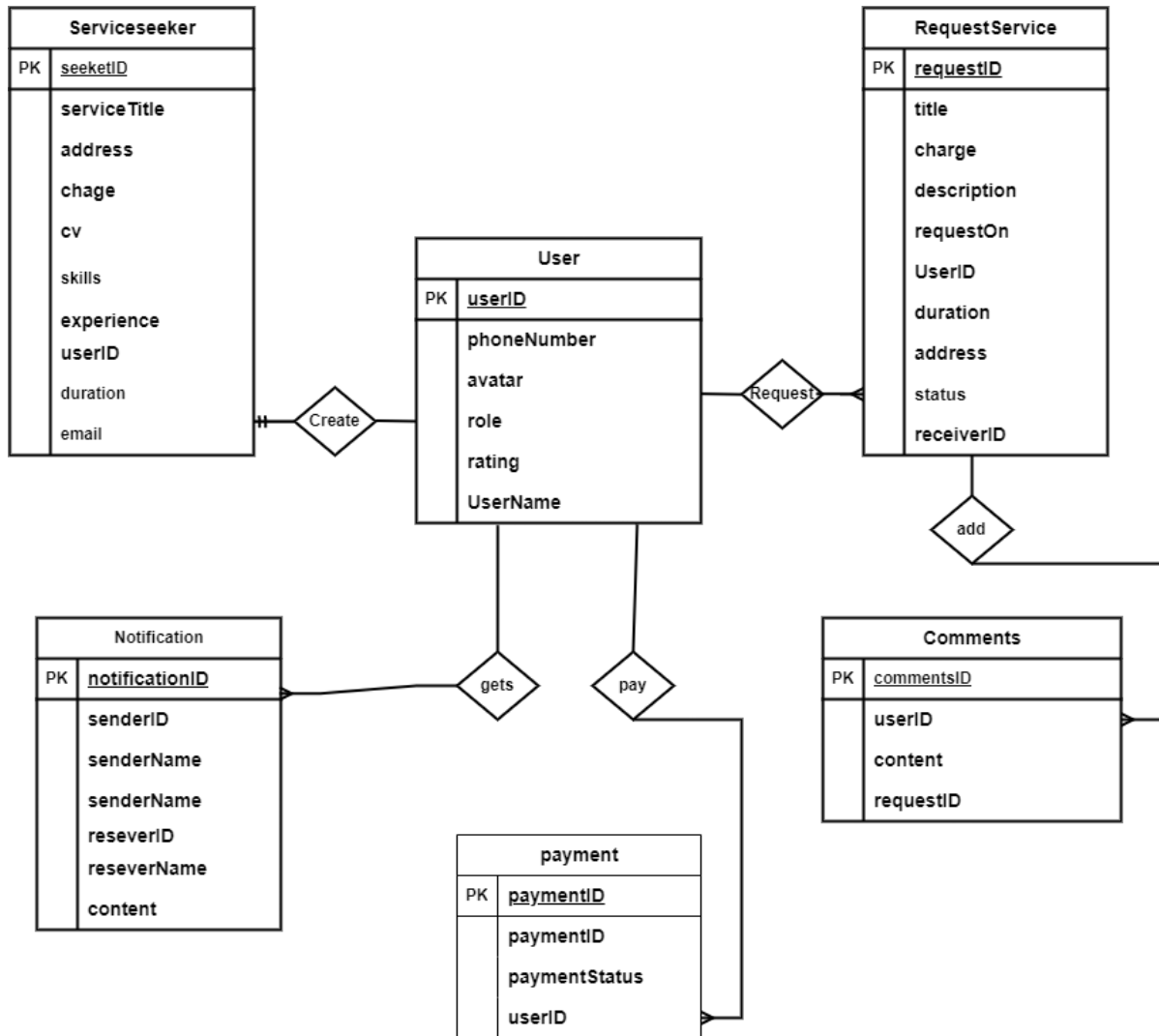


**Figure 4: Entity Relationship Diagram(sayogi)**

In the above ER Diagram, we can see clearly all the relation between the entities. There are four total entities in this project and they have their respective attributes. Here admin has all authorize to manage the User, he can delete user, add user. And User can create service Request, make profile as a service Provider, Request to hire, get Notification, and add comments.

## 3.1.4 Process Modeling (DFD)

Data flow diagram is graphical representation of flow of data in an information system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination.

**Context diagram**

A context diagram, also known as a level 0 diagram, is a visual representation of a system or process that shows the interactions between the system and external entities. It provides an overview of the system's boundaries, the entities it interacts with, and the flow of data between them.
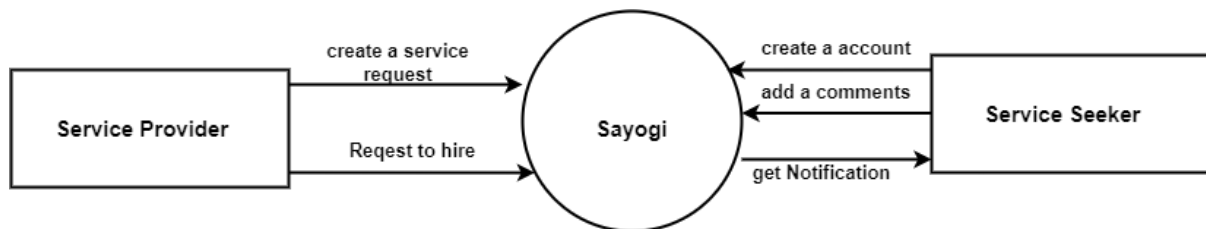


**Figure 5: Context diagram**

**1-level DFD**

DFD Level 1 provides a more detailed breakout of pieces of the Context Level Diagram. This DFD describes main functions carried out by the system, as we break down the high-level process of the Context Diagram into its sub-processes.
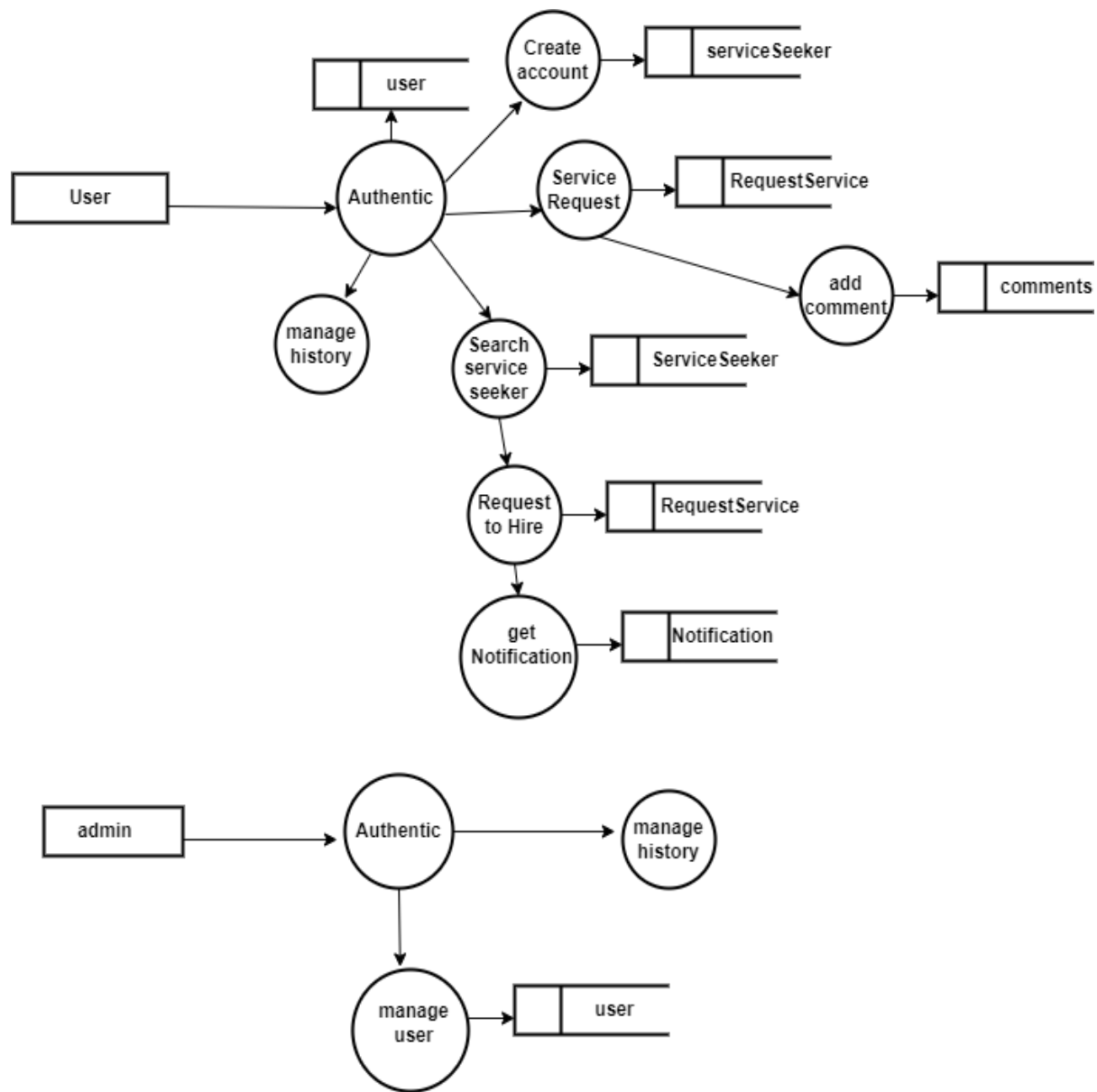
**Figure 6: 1 level DFD**

## 3.2 System Design

System Design involves creating a detailed blueprint or plan for the development of a new system or the enhancement of an existing one. It encompasses defining system components, specifying their functionalities, and outlining their interactions to achieve specific goals. This phase bridges the conceptualization of a project with its actual implementation, detailing the structure, architecture, and interfaces. The design process considers factors like scalability, flexibility, and user experience, ensuring the system meets both current and future requirements. The ultimate goal of System Design is to provide a comprehensive guide for developers, laying the foundation for efficient and effective system development.
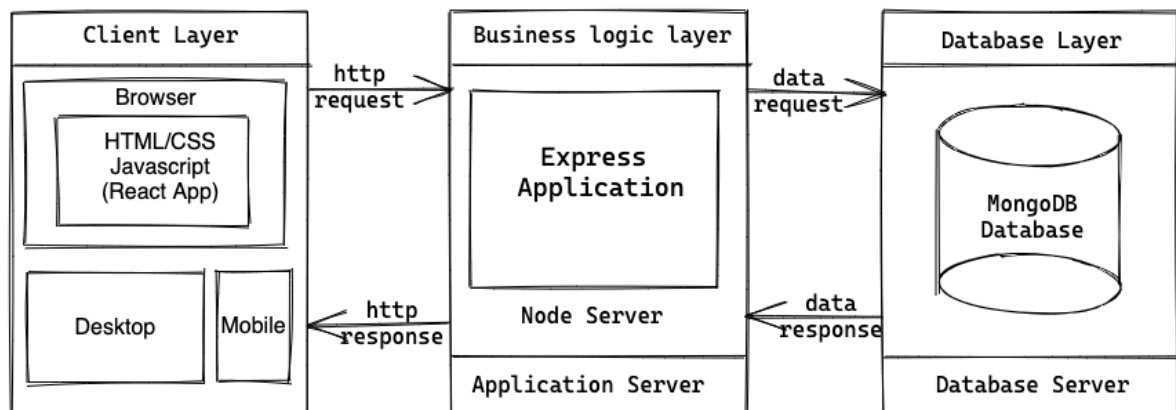
### 3.2.1 Architectural design



**Figure 7: System Architectural Design**

The next phase of design will be planning the architecture of the application. Sayogi makes the use of internet and computer peripherals. This system is supported by any device that can access the internet. The system architecture is clearly displayed in the figure above.

### 3.2.2 Database Schema

A database schema is the skeleton structure that represents the logical view of the entire database. It defines how the data is organized in a system.
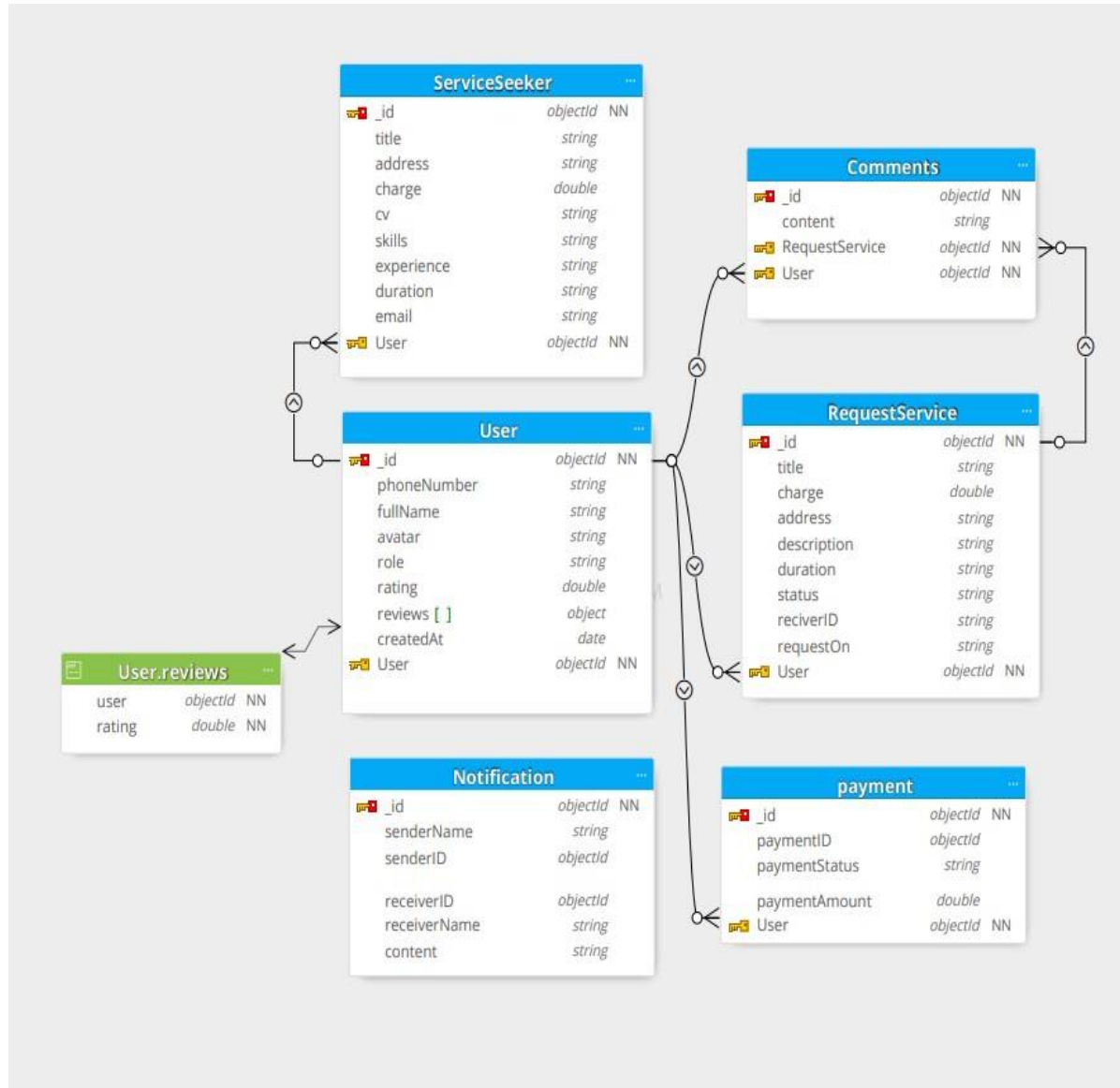
**Figure 8: Database Schema Design of Sayogi**

### 3.2.3 Interface design (UI/UX)

Before implementing the actual design of the project, a few user interface designs are constructed to visualize the user interaction with the system as they browse for view, Home, Authentication, Service Request, History. The user interface design will closely follow our Functional Decomposition Diagram showing the initial designs of the web pages.
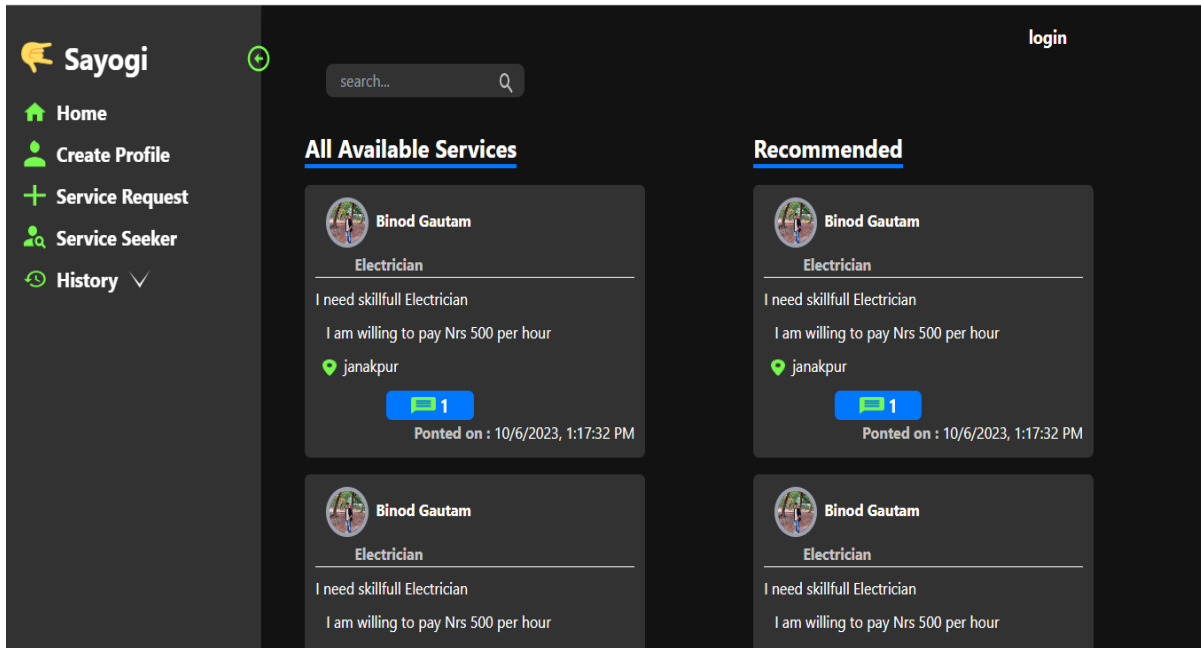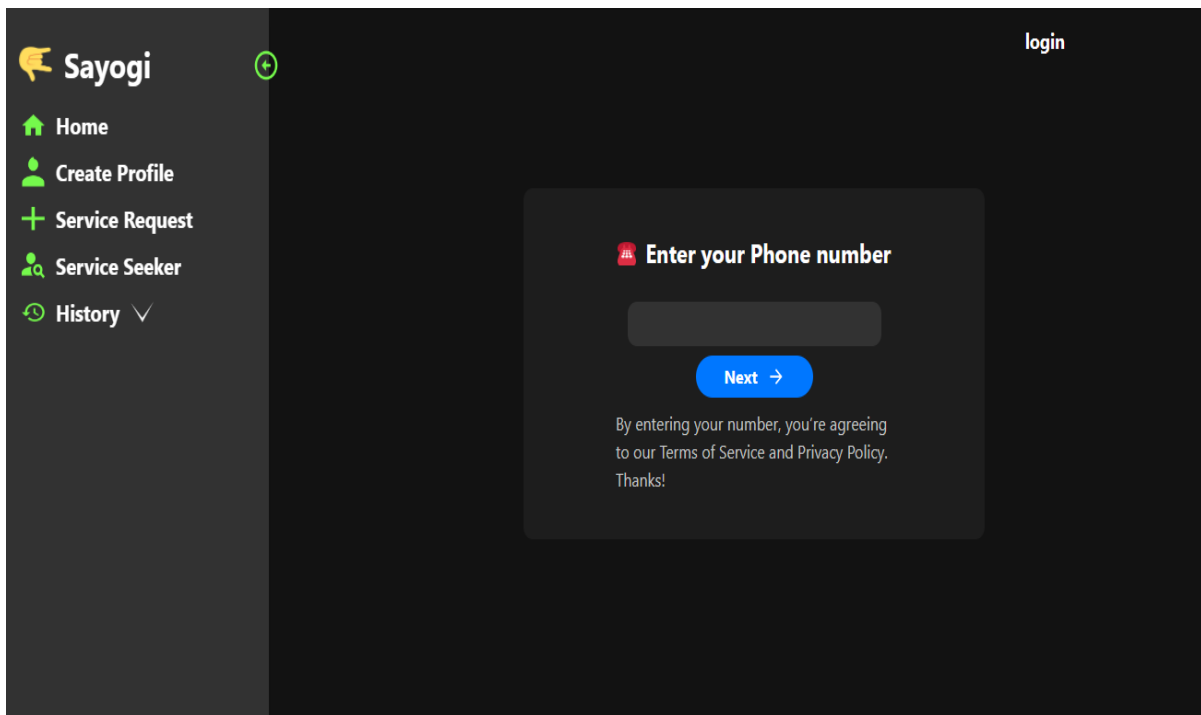
**Figure 9: Home page of Sayogi**

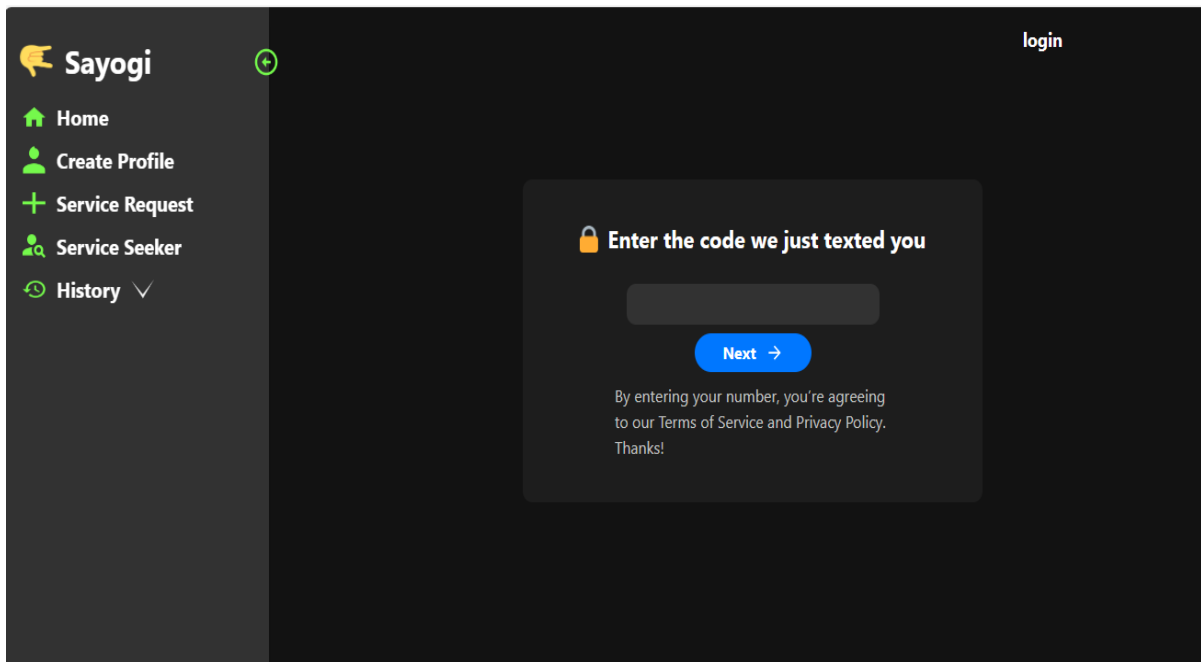

**Figure 10: Authentication 1 page of Sayogi**

**Figure 11: Authentication 2 page of Sayogi**
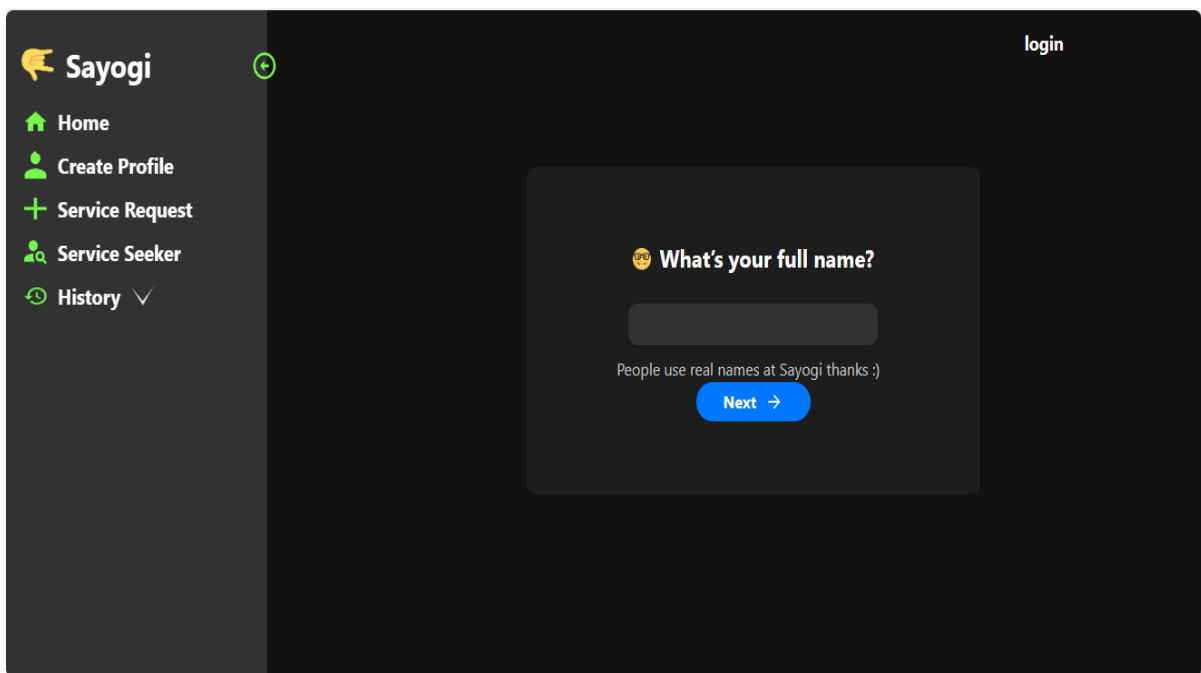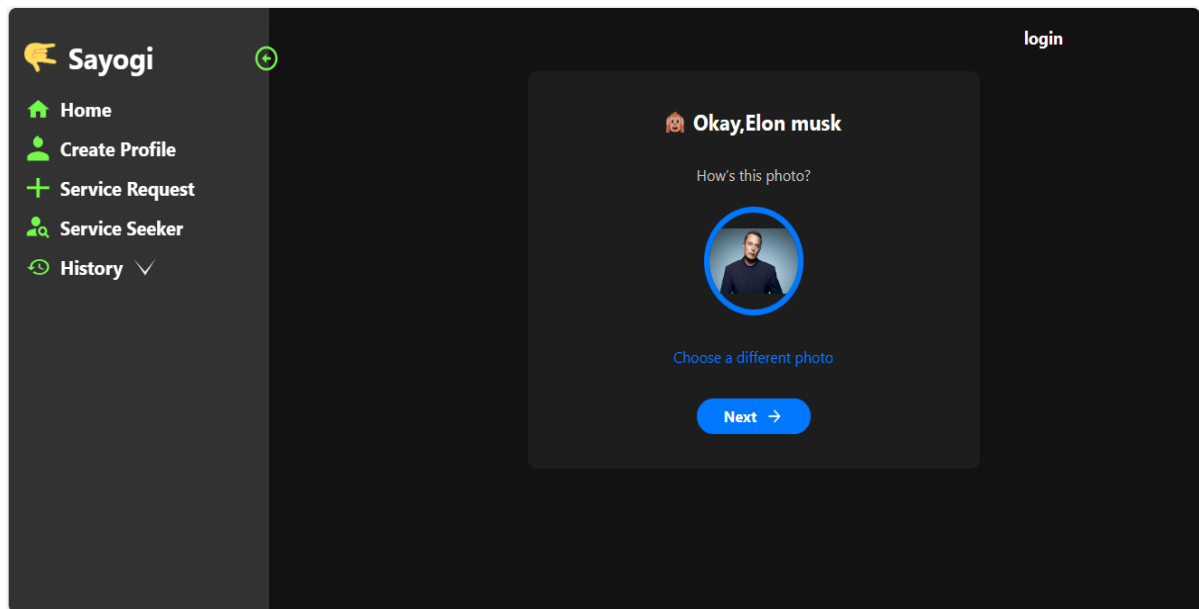


**Figure 12: Authentication 3 page of Sayogi**

**Figure 13: Authentication 4 page of Sayogi**



**Figure 14: Create Service seeker profile page of Sayogi**

**Figure 15: Service Create page of Sayogi**



**Figure 16: Service Seeker detail page of Sayogi**
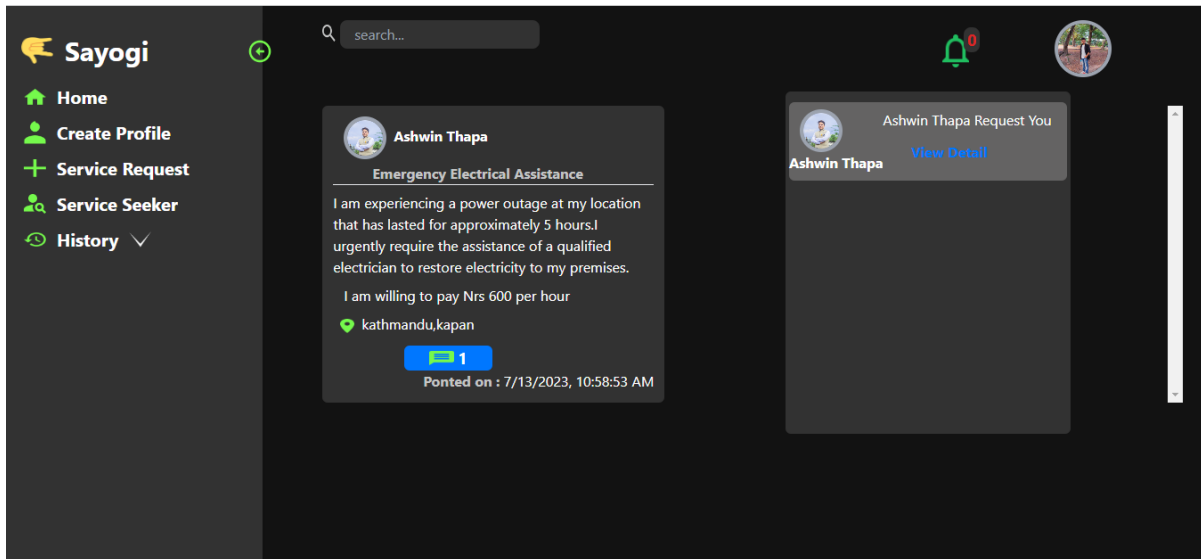
**Figure 17: Notification  page of Sayogi**
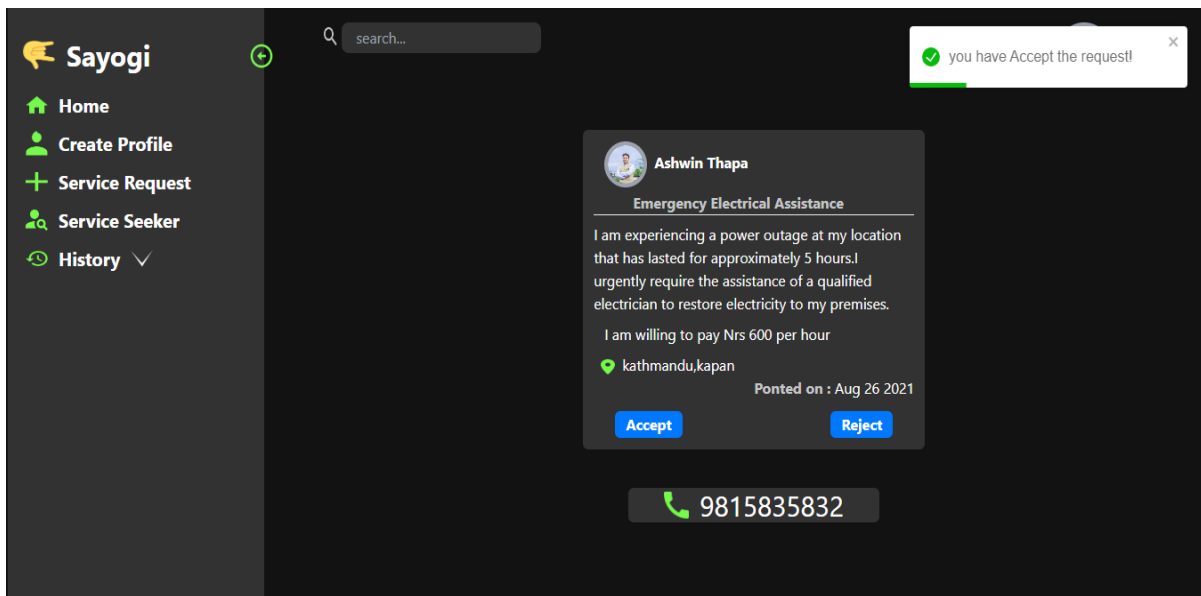


**Figure 18: Service Detail page of Sayogi**
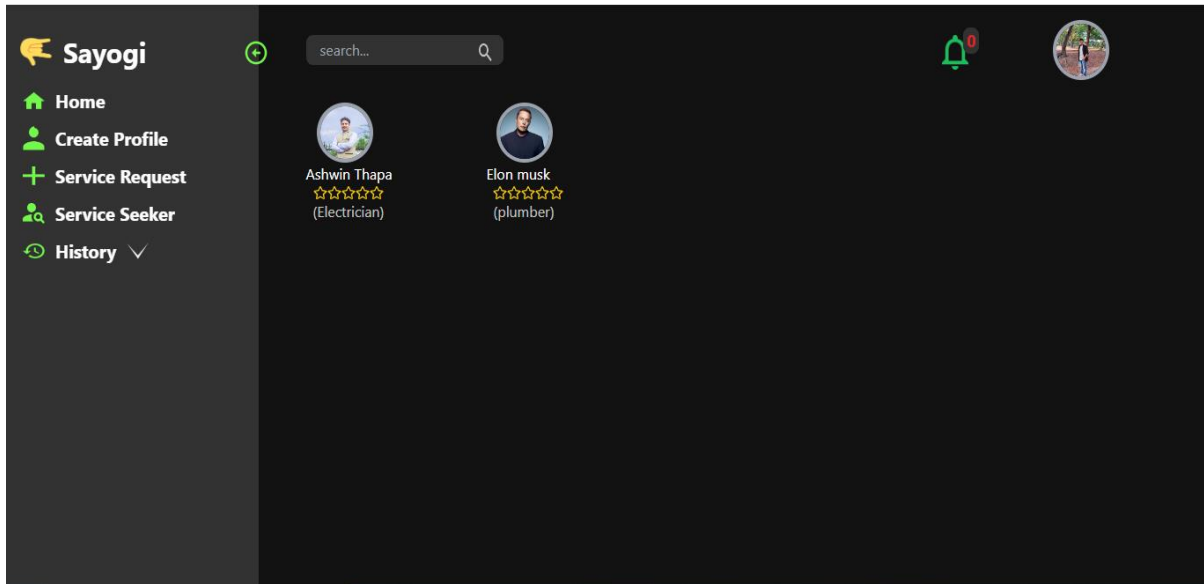
**Figure 19: All service Seeker search page of Sayogi**
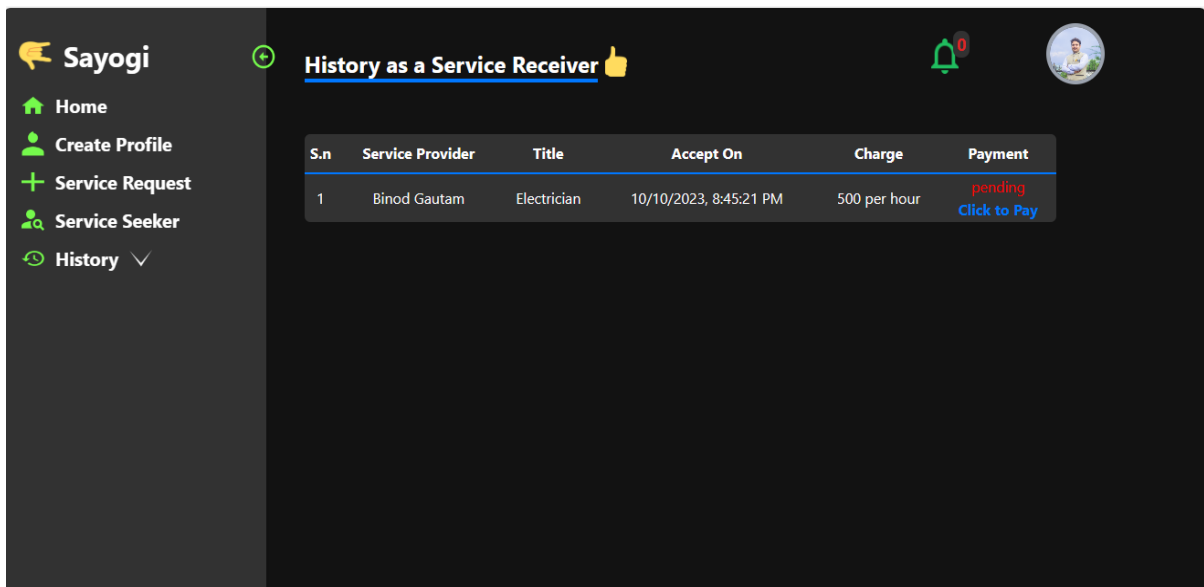


**Figure 20: History as a service Receiver  page of Sayogi**

**Figure 21: History as a Provider page of Sayogi**



**Figure 22: Payment page of Sayogi**
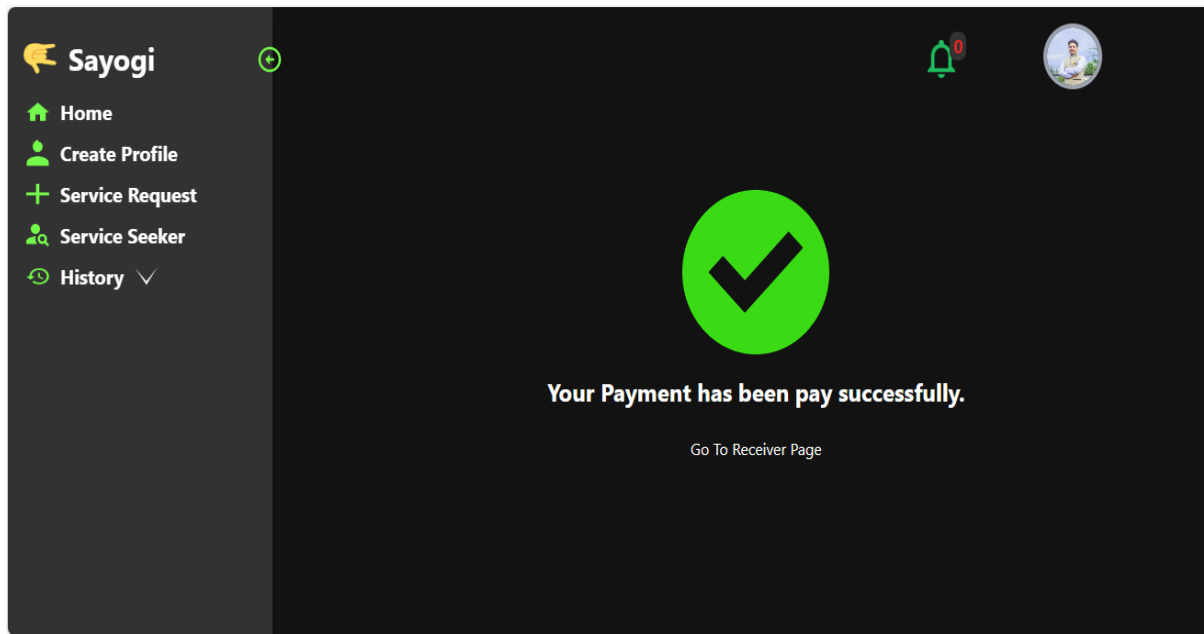
**Figure 23: Payment sucess page of Sayogi**
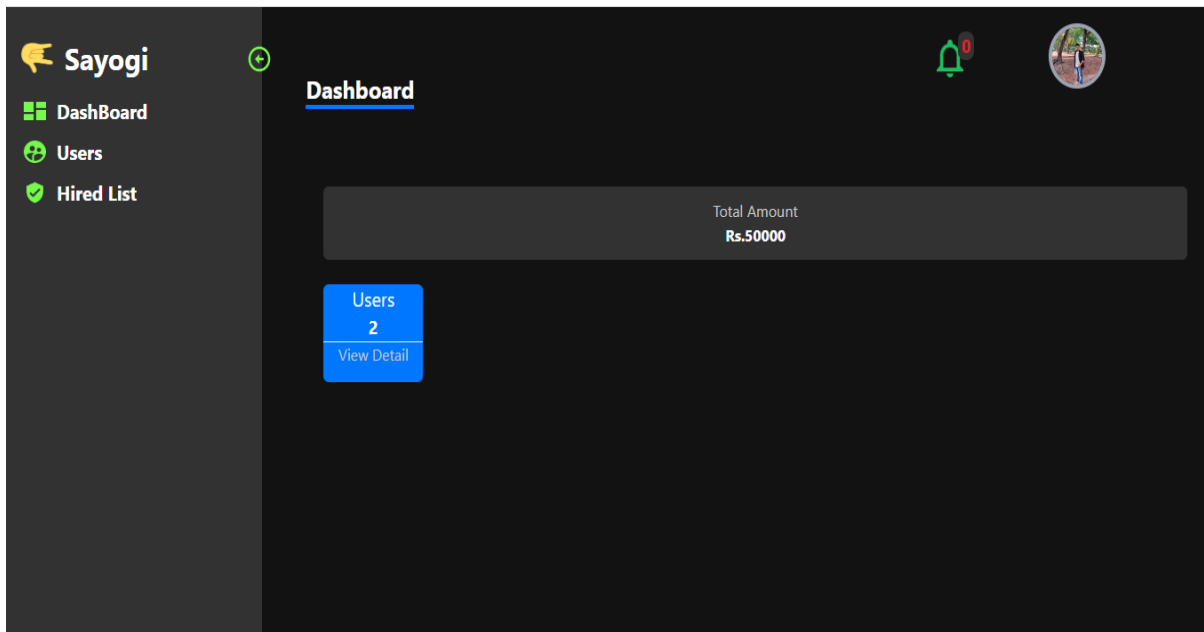


**Figure 24: Admin Dashboard page of Sayogi**
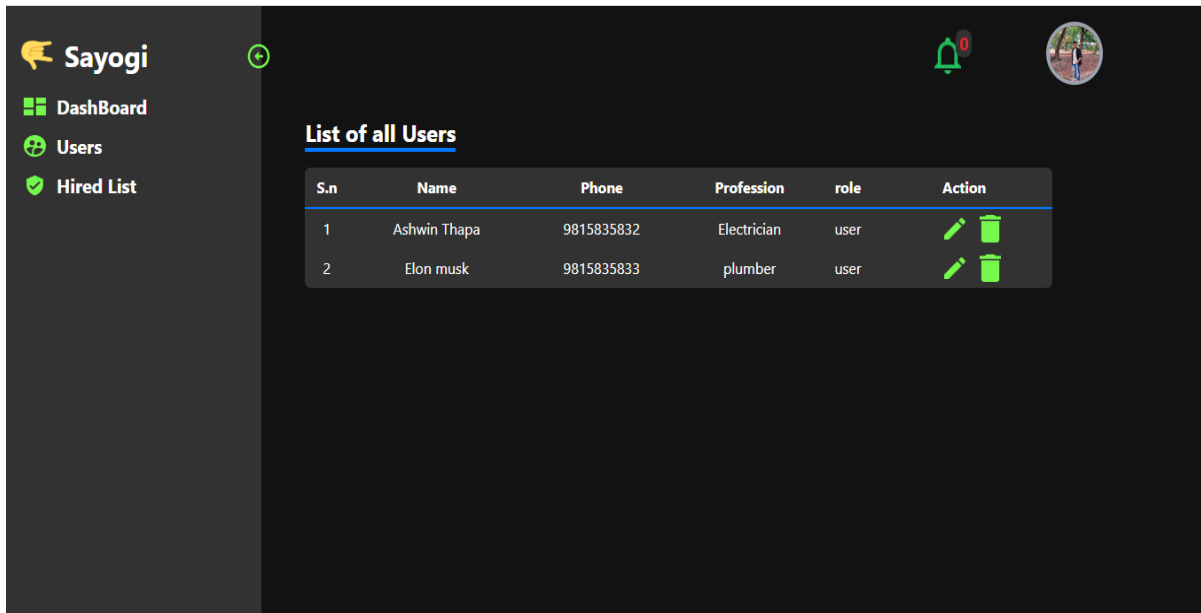
**Figure 25: All Hired User List  page of Sayogi**
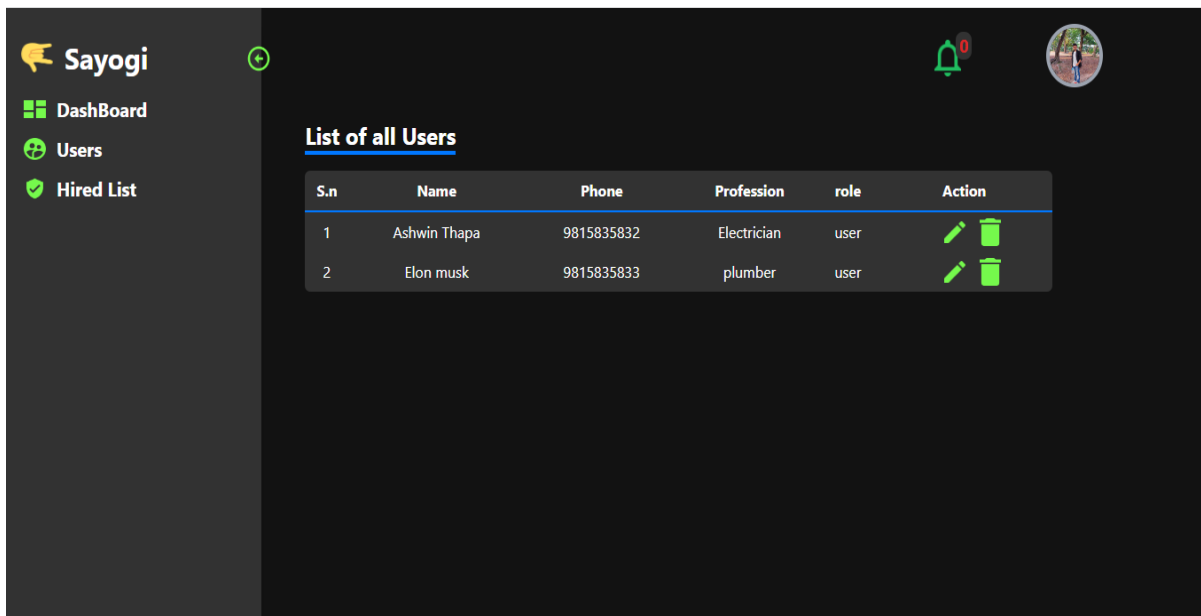


**Figure 26: List of all page of Sayogi**

## 3.3 Algorithm details

This algorithm serves as a backend service for suggesting service requests within the Sayogi project, allowing users to specify preferences such as title and address. In practical terms, it's a key component of the Sayogi system where service seekers inp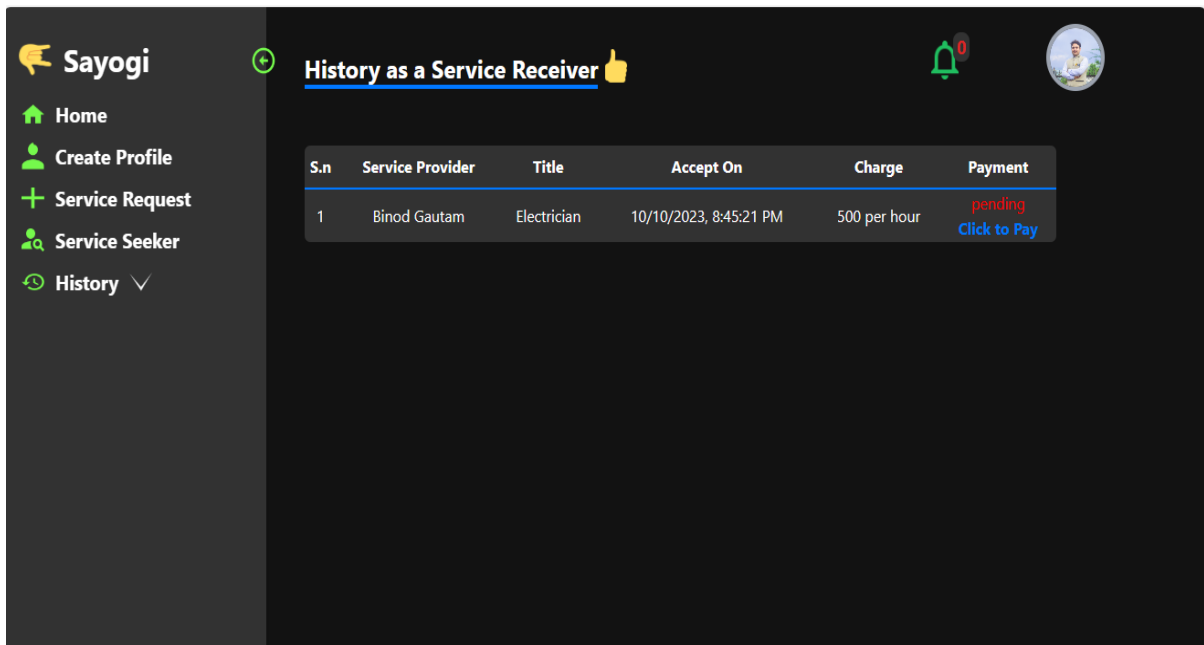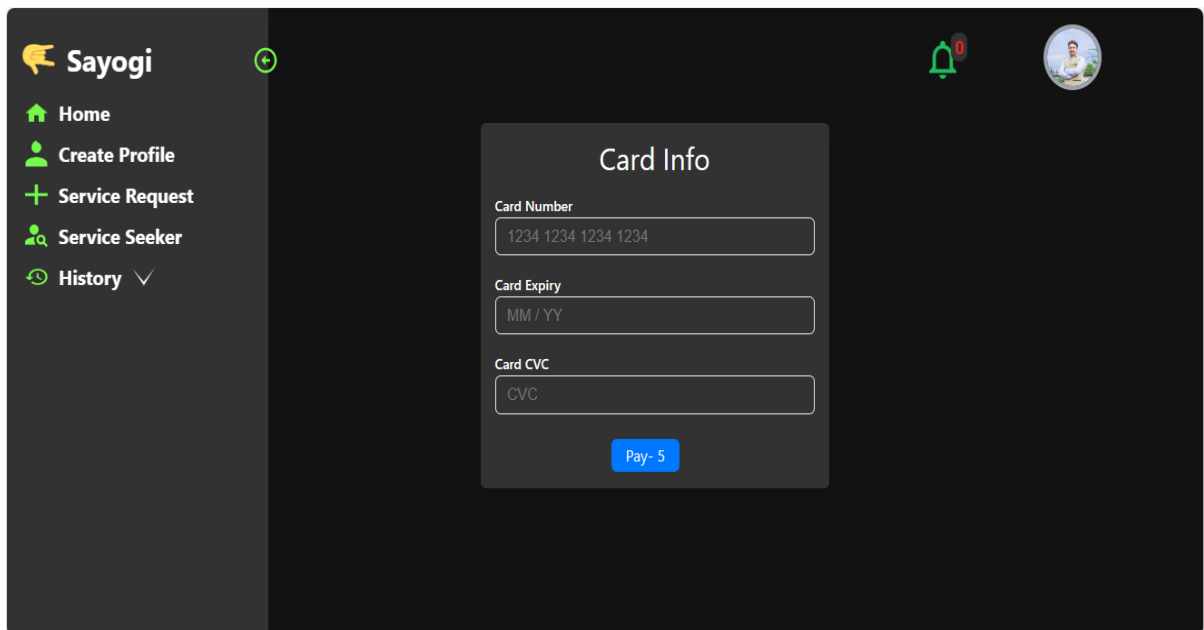ut their criteria, and the algorithm retrieves relevant service requests from a database. The filter object plays a crucial role in constructing MongoDB queries, facilitating dynamic and flexible matching. Whether a user specifies a title or address, the algorithm utilizes regular expressions for case-insensitive matching. Notably, it gracefully handles cases where users don't provide specific criteria, ensuring the return of service requests even if only one parameter is specified. The asynchronous nature of the function implies a focus on efficiency, preventing the service from blocking while waiting for the database query to complete.

In essence, this algorithm seamlessly integrates into the Sayogi project, streamlining the process of connecting service seekers with tailored suggestions. By allowing users to refine their preferences, it enhances the overall user experience, and the backend service ensures that the suggestions provided align precisely with the specified criteria.

# Chapter 4: Implementation and Testing

## 4.1 Implementation

After designing the system, the only thing that needs to be done is implement it so that we can release it as per the user satisfaction. Implementing the system requires a lot of resources and explanation which will not be completely explained in this report; however, some major aspects of the system are described below:

### 4.1.1 Tools used

**MongoDB:**

It's a NoSQL database that seamlessly integrates with Node.js, providing a flexible and scalable solution. In Sayogi, MongoDB will store and manage service requests, ensuring quick and efficient retrieval of data.

**Express.js:**

Express.js facilitates the development of the backend server, handling HTTP requests and enabling the creation of robust APIs. It plays a pivotal role in implementing the service suggestion algorithm, ensuring seamless communication between the frontend and the MongoDB database.

**React:**

React forms the frontend of the Sayogi project, offering a declarative and component-based approach to building user interfaces. With React, the frontend can dynamically update based on user interactions, such as refining preferences or inputting information. It ensures a responsive and engaging user experience.

**Node.js:**

Node.js serves as the runtime environment for the backend, allowing JavaScript to be used on both the frontend and backend. This unification streamlines development, as developers can use the same language throughout the entire stack, enhancing code reusability.

**Tailwind CSS:**

Tailwind CSS is chosen for the project's styling needs. Its utility-first approach and customizable nature make it an excellent fit for designing the Sayogi frontend. With Tailwind, developers can create a visually appealing and responsive user interface without sacrificing flexibility.

**Visual Studio Code**

We use vs code as a code editor because this support hundreds of languages, also it is the most commonly used code editor. VS code helps we be instantly productive with syntax highlighting, bracket matching, auto-indentation, box -selection, snipping and more, so we think VS code is the best for our project.

**draw.io**

Draw.io is proprietary software for making diagrams and charts. In our project we use draw.io to make architectural diagram, ER diagram, use case diagram and DFD.

**4.1.2 Software Development Model**

Software development model is the series of processes used in software development. The commonly used methodologies include agile development methodology, waterfall method and rapid application development. There are few other methodologies depending upon the nature and objective of the software. In developing our entire system, we used the Waterfall Development Model. This was the most suitable model for our system. In addition, this model is very simple and easy to understand. Each phase must be completed before the new phases start, so there is no overlapping in the phases. The different development cycle was broken down into a sequence of processes and thus the development of Sayogi was progressed. The following illustration is a representation of different phases of the waterfall model:
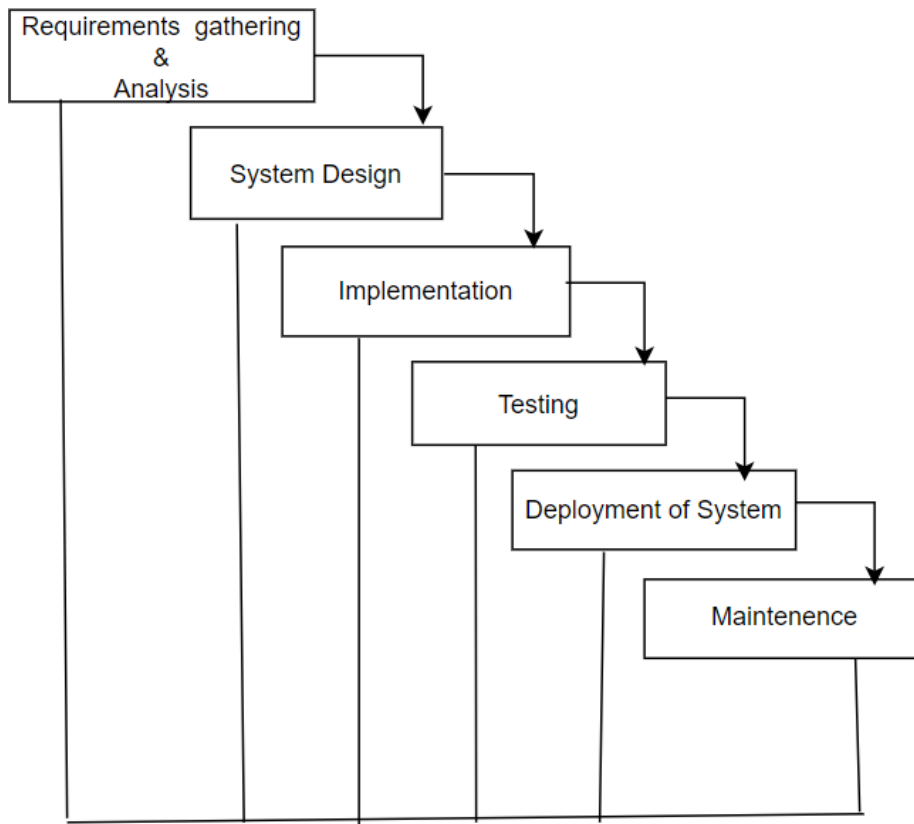
**Figure 27: Waterfall Model**

The above figure shows us the implementation of Waterfall Model. Being the most suitable model to use in our application we followed its approach where we would only continue to next module after the completion of previous module.

The following is the representation of different stages of waterfall model:

i.  **Requirement Gathering and Analysis:** Before developing the system, it is crucial for requirement collection, and analysis. This phase was used to determine all such requirements that helped to make our product complete. This phase involves engaging with stakeholders, including service seekers and providers, to comprehensively understand and document their input on essential features and functionalities. The defined project objectives and goals are documented in a detailed requirements document, which undergoes thorough analysis and prioritization.

ii. **System Design:** In this phase, the framework of the proposed system was designed from the requirements which were addressed in the data collection phase. This stage will be useful in specifying the needed hardware and system requirements, including the system structure. We design the overall working flow of the system along with designing system architecture, ER diagram, Data Flow Diagram and Use Case Diagram. The whole system is divided into the different components and modules ie client side module, admin side module. The data flow of the system is explained perfectly in this phase.

iii. **Implementation:** Our implementation phase started with the completion of our designing phase. In this stage, as per the design we converted it into the runnable code. We had chosen a Node js as a javascript run time. All the backend logic is implemented javascript and user interface is implemented using React js In this way we implemented our system using Mongodb, tailwind css, node js ,Express js and React js.

iv. **Testing:** After the implementation phase is totally finished, another phase is testing. In this phase we tested the each functionality of the system. If the system does not work as per the function defined above the system will fail and again the error must be fixed. For-example the service provider Request to hire to the service Seeker, in this case if the Request is not placed it seems to be error. In this way we tested the each and every thing in the system then only we moved in another phase.

v. **Deployment of System:** Once the system is tested carefully, product is deployed in the customer environment or released in the market. Thus, in this phase product is given to user to use. We ourself use this system as a user and we consider that admin dashboard is not good. So again we moved into the another phase.

**vi.** **Maintenance:** After the product is deployed there are some issues which come up in the client environment. To fix those issues, patches are released. Also, to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

## 4.2 Testing

Testing is the major task in software development. We are not confirmed that our software is ready to use unless the testing is done. There are various types of testing in software development.

In the testing phase of Sayogi, various approaches are employed to ensure the application's reliability and functionality. This includes unit testing for individual components, integration testing to validate communication between frontend and backend, end-to-end testing for user scenarios, performance testing to optimize load times, and user acceptance testing involving real users to validate real-world usability. Continuous feedback and iterative testing processes aim to identify and rectify issues promptly, ensuring the development of a robust and user-friendly platform.

### 4.2.1 Manual Testing

**Table 1: Test Case for Login**

| ID | Test Case Description | Test Data | Expected Result | Actual Result | Pass/Fail |
|----|----------------------|-----------|-----------------|---------------|-----------|
| TC_01 | User enter invalid phone number | 981583583 | Phone number must be 10 digit | As expected | pass |
| TC_02 | User enter invalid otp number | 034 | Otp must be 4 digit | As expected | pass |
| TC_03 | User enter valid otp and phone number | Phone: 9815835831 Otp: 1234 | Logged into dashboard | As expected | pass |

**Table 2: Test Case for Creating Service Seeker profile**

| ID | Test Case Description | Test Data | Expected Result | Actual Result | Pass/Fail |
|---|---|---|---|---|---|
| TC_04 | User enter all Required data | Service Title: Electrician Experience: 3 years Address: Kathmandu,kepan Expected Charge: 500 per hr | Profile create Successfully. | As expected | pass |
| TC_05 | User enter leave the field empty | Service Title: Electrician Experience: 3 years Address: Kathmandu,kapan Expected Charge: | All the filed must be filled | As expected | pass |

# Chapter 5: Conclusion and Future Recommendations

## 5.1 Conclusion

In concluding the Sayogi project, we've successfully brought to life a user-friendly platform that effectively bridges the gap between service seekers and providers. Leveraging the MERN stack, Tailwind CSS, and Mongoose as the ODM tool, we've ensured a scalable, responsive, and intuitively designed application that aligns with contemporary web development standards.

The thorough testing processes, from unit testing to system testing, coupled with continuous collaboration with stakeholders, have refined Sayogi into a robust and secure platform. As we move forward, Sayogi stands poised to make a significant impact in the service-oriented marketplace, offering a seamless experience for users in connecting with the right service providers. Ongoing refinements and enhancements will further solidify Sayogi position as an innovative and efficient solution for diverse service needs.

## 5.2 Lesson Learnt/Outcome

In the Sayogi project, we made sure our focus was on what users needed, following a waterfall development plan. The tools we picked—MERN stack, Tailwind CSS, and Mongoose—worked really well, showing how important it is to choose the right technologies. We did thorough testing, checking each part and the whole system, to make sure everything worked well. The project worked out because we talked openly, adapted to changes, and always aimed to make things better.

## 5.3 Future Recommendations

Looking ahead, Sayogi success can be further amplified by prioritizing user feedback integration and implementing advanced analytics tools to gain deeper insights into user behavior. Proactive planning for scalability, exploration of AI and automation, and a steadfast commitment to cyber security measures will fortify the platform's robustness.

Additionally, considering the widespread use of mobile devices, the development of a dedicated mobile application could enhance accessibility. To foster a vibrant user community, features like discussion forums and community events can be integrated.

Ensuring compliance with accessibility standards, planning for internationalization, and providing regular training for service providers will contribute to the platform's global appeal and longevity. These future recommendations align with Sayogi commitment to continuous improvement and user satisfaction.

# References

https://www.freelancer.com/.

https://www.fiverr.com/