SKILL SWAP PLATFORM

Project Report Submitted in Partial Fulfilment of the Requirements for

the Degree of

**Masters**

***Of***

**Computer Applications**

*Submitted by*

Pratik Prakash Sharma :- 1000021363 Mudit Chauhan :- 1000019824 Abhishek Semwal :- 1000019759

***Under the Supervision of***

Mr. Sanjay Kumar



# DIT UNIVERSITY, DEHRADUN

(State Private University through State Legislature Act No. 10 of 2013 of Uttarakhand and approved by UGC)

**Mussoorie Diversion Road, Dehradun, Uttarakhand – 248009, India.**

# May, 2025

**DECLARATION**

We declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the

original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed. The plagiarism check report is attached at the end of this document.

Name of the Student: Pratik Prakash Sharma Name of the Student: Mudit Chauhan Name of the Student: Abhishek Semwal

Signature and Date Signature and Date Signature and Date

# ACKNOWLEDGEMENT

This project has required a lot of effort from us. It would not have been feasible without the generous support and assistance of many people. We would like to express our heartfelt

gratitude to every one of them.

First and foremost, we are grateful to **Mr. Sanjay Kumar** , our project guide, for his

guidance in advancing our study. We are grateful to him for devoting his time and attention to us, as well as for providing us with a methodical approach to completing our job on time.

Then we are thankful to all members of this project, without their support and handwork’s this could never be possible to make any progress in this project.

**Prateek Prakash Sharma**

## Abhishek Semwal Mudit Chauhan

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S.NO.** | **TOPIC** | **PAGE NO.** |
| 1. | Introduction | 1-4 |
| 2. | Objectives | 5-6 |
| 3. | Tools and technology stack | 7-8 |
| 4. | System architecture | 9-11 |
| 5. | Module description | 12-14 |
| 6. | Data flow diagram | 15-16 |
| 7. | Use case diagram | 17-18 |
| 8. | Implementation and Screenshots | 19-21 |
| 9. | Conclusion | 22 |
| 10. | Scope for future work | 23-24 |
| 11. | References | 25-26 |

# LIST OF FIGURES

FIGURE NAME AND NO. PAGE NO.

1. Overview diagram 9
   1. DFD (Level 0) 15
   2. DFD (Level 1) 16
   3. Use case diagram 17
   4. Home page 19
   5. Login and sign up 19
   6. Home page after login 20
   7. Profile page 20
   8. Skills section 20
   9. Swap request page 21
   10. Request in profile 21
   11. Connecting through whatsapp 21

# ABSTRACT

In today’s fast-paced digital world, skills are currency. However, not everyone has equal access to formal learning resources. The **Skill Swap Platform** aims to bridge this gap by enabling individuals to exchange skills directly with others in a collaborative and cost- effective manner. This web-based application allows users to register, list their skills, and

request to swap them with others who have complementary expertise. Whether it’s graphic design for coding, language learning for digital marketing, or photography for writing, the platform fosters a community-driven learning environment.

The project is built using the MERN stack (MongoDB, Express.js, React, Node.js), ensuring a modern, scalable, and responsive interface. Key features include secure user

authentication, real-time swap request notifications using WebSockets, profile management, messaging, scheduling of skill exchange sessions, and feedback/rating systems. The platform also integrates social features and gamification to increase user engagement.

By focusing on peer-to-peer interaction and mutual growth, the Skill Swap Platform promotes lifelong learning, networking, and community development—making it not just a tool, but a movement toward democratized skill development.

# CHAPTER 1

**INTRODUCTION**

## Introduction

**Skill Swap Platform**

In today’s interconnected digital world, the way people acquire, share, and apply skills is evolving rapidly. Traditional education and professional training often come with high costs, geographical limitations, and inflexible structures, leaving many potential learners and teachers underserved. Recognizing the growing demand for accessible and community- driven learning, our team conceptualized and developed the Skill Swap Platform—a versatile, user-centric application designed to facilitate peer-to-peer skill exchange on a local, national, and even global scale.

Originally envisioned as a simple tool for connecting individuals within small communities, the platform has expanded into a full-fledged ecosystem accessible via web and mobile devices. It empowers users to list the skills they can teach, specify the skills they want to learn, and find compatible matches through intelligent recommendation systems. Secure messaging, session scheduling, and a built-in feedback and rating mechanism ensure a trustworthy and seamless experience for all participants.

One of the platform’s defining features is its scalability and adaptability. Whether users are located in a small town, a bustling city, or across different countries, Skill Swap enables them to connect based on shared learning goals. Real-time updates, multi-language support, and flexible time zone scheduling make the platform truly global. Furthermore, administrators have access to robust analytics tools to monitor platform trends, popular skill exchanges, and user activity, supporting continuous improvement and expansion.

By harnessing modern development technologies and focusing on community empowerment, the Skill Swap Platform serves as a dynamic solution for lifelong learning, skill diversification, and global networking, breaking down traditional barriers to education and personal growth.

## Key features of Skill Swap Platform

* + 1. **Skill Listings**
       - Users can list the skills they offer and the skills they seek in exchange.

## Smart Matching

* + - * Intelligent algorithms suggest potential swap partners based on skill compatibility, availability, and location (if relevant).

## Profile Verification

* + - * Secure user verification through ID checks and reviews to build a trusted community.

## In-App Messaging

* + - * Chat directly with potential partners to negotiate and finalize skill exchange agreements.

## Skill Portfolio

* + - * Users can build and showcase a portfolio of completed swaps, endorsements, and testimonials.

## Virtual and In-Person Options

* + - * Support for both online sessions (video/audio calls) and local meetups, depending on user preference.

Overall, the Skill Swap Platform reimagines how people learn and share abilities in a collaborative community. By connecting individuals through mutual skill exchanges, digitizing matchmaking, and fostering trust with verified profiles and feedback, the platform creates a dynamic, accessible, and empowering ecosystem for personal and professional growth.

## Project Description

The **Skill Swap Platform** is a full-fledged, user-driven web application that facilitates the mutual exchange of skills among users across different domains. The idea behind the platform is simple yet impactful: instead of paying for learning a skill, a user can offer one of their own skills in return. This approach fosters a sense of community, promotes accessible education, and reduces financial barriers to learning.

At its core, the platform provides the following functionality:

* **User Registration and Authentication**: New users can register by providing a username, password, email, mobile number, and other profile details. Passwords are secured, and validations ensure that usernames and mobile numbers are unique. Users can also log in, log out, and reset or change their passwords.
* **Skill Management**: Users can add multiple skills to their profile, including skill name, proficiency level, and a brief description. These skills are visible to other users, making it easier to find potential swap partners.
* **Skill Search and Filtering**: The homepage includes a dynamic search feature that lets users browse and filter skills based on keywords, categories, or user ratings. This enhances discoverability and simplifies the match-making process.
* **Swap Request System**: Once a user finds a skill they are interested in, they can click the **"Swap"** button to send a request, including which of their own skills they are offering in return. This initiates the skill exchange process.
* **Real-Time Notifications**: The platform uses WebSockets (e.g., with Socket.IO) to provide real-time updates. Users are instantly notified when they receive a swap request or when their request is accepted or rejected.
* **Discussion Dashboard**: When a swap is accepted, a dedicated dashboard opens up where users can chat, share resources (e.g., links, documents), and coordinate the swap. The dashboard also includes scheduling features, allowing users to fix meetings via integrated services such as WhatsApp or Microsoft Teams.
* **Ratings and Reviews**: After each swap, users can leave feedback and ratings, helping maintain trust and quality within the community.
* **Gamification and Engagement**: To increase user participation, the platform incorporates gamification features such as badges, experience points, and levels based on successful swaps and user activity.
* **Admin Panel**: A backend admin interface allows administrators to manage users, monitor activities, handle disputes, and ensure the platform remains secure and user- friendly.
* **Security and Privacy**: The system uses secure password hashing, input validation, and session management to protect user data and ensure privacy.

## Technology Stack:

* + **Frontend**: React.js with Tailwind CSS for a responsive and modern UI.
  + **Backend**: Node.js with Express.js for scalable RESTful APIs.
  + **Database**: MongoDB for storing user data, skills, swap requests, and messages.
  + **Others**: Socket.IO for real-time communication, third-party API integrations for messaging/scheduling, and JWT or session-based authentication for security.

# CHAPTER – 2 OBJECTIVES

## OBJECTIVES

The primary objective of the **Skill Swap Platform** is to create a user-friendly, secure, and interactive web application that enables users to exchange skills without monetary transactions. The platform promotes peer-to-peer learning and fosters a global knowledge- sharing community. The specific objectives of the project include:

## Main Objectives

* + - To design and develop a platform where users can **register, manage their profiles**, and **list their skills**.
    - To allow users to **search for and request a swap** with others who have the desired skills.
    - To implement a **mutual skill exchange system** that encourages collaboration and self- improvement.
    - To support **real-time notifications and updates** using WebSockets for efficient communication.
    - To provide a **dedicated dashboard** where users can chat, schedule meetings, and share learning resources.
    - To enable users to **rate and review** each other after a skill swap, building trust and reputation.

## Technical Objectives

* + - To use the **MERN (MongoDB, Express.js, React.js, Node.js)** stack for robust and scalable full-stack development.
    - To implement **secure user authentication and validation**, including unique usernames and mobile numbers.
    - To store and manage user data, skills, and swap requests effectively using **MongoDB**.
    - To integrate **external tools/APIs** such as WhatsApp and Microsoft Teams for real-time communication and scheduling.
    - To implement **admin control functionalities** for monitoring and maintaining the platform’s integrity.
    - To design a **responsive, modern UI** using React and Tailwind CSS for seamless user experience across devices.

## Future-Oriented Objectives

* + - To incorporate **AI-powered skill matching** and personalized recommendations.
    - To add **mobile application support** for broader accessibility.
    - To introduce **gamification elements** like badges, achievements, and levels to boost engagement.
    - To expand the platform to support **multiple languages** for global reach.
    - To ensure **data privacy, security, and scalability** as the platform grows.

# CHAPTER-3

**TOOLS AND TECHNOLOGY STACK**

## TOOLS AND TECHNOLOGY STACK

The Skill Swap Platform is developed using modern web technologies to ensure high performance, scalability, security, and user-friendly design. Below is a breakdown of the tools, frameworks, and technologies used throughout the development of the project:

* 1. Frontend Technologies
     + React.js: A powerful JavaScript library used for building the user interface with reusable components and a dynamic, responsive experience.
     + Tailwind CSS: A utility-first CSS framework for designing modern and responsive layouts quickly.
     + HTML5 & CSS3: Standard web technologies for structuring and styling the frontend.
     + JavaScript (ES6+): For client-side scripting, interactivity, and logic.
  2. Backend Technologies
     + Node.js: A JavaScript runtime environment used to build the server-side logic of the application.
     + Express.js: A lightweight and flexible Node.js framework used to build RESTful APIs for handling requests and responses.
  3. Database
     + MongoDB: A NoSQL database used to store user data, skills, swap requests, messages, and ratings in a flexible and scalable format.
  4. Real-Time Communication
     + Socket.IO: Enables real-time, bidirectional communication between the server and client for instant swap request notifications and chat features.
  5. Authentication and Security
     + bcrypt.js: For hashing passwords securely before storing them in the database.
     + express-session / JWT (JSON Web Tokens): For managing secure sessions and user authentication.
     + Validator.js: For input validation to ensure data integrity.
  6. APIs and Integrations
     + WhatsApp API / Microsoft Teams: Integrated for real-time communication and scheduling of virtual meetings.
  7. Development and Deployment Tools
     + Visual Studio Code: Primary code editor used for writing and managing code efficiently.
     + Postman: API testing tool to verify backend endpoints during development.
     + Git & GitHub: Version control system used to track changes and collaborate.
     + Vercel / Netlify: Used for frontend deployment, offering CI/CD features.
     + MongoDB Atlas: Used for backend and database deployment in the cloud.
  8. Browser Compatibility and Testing
     + Chrome Developer Tools: For debugging and performance testing.
     + Responsive Design Mode: To ensure the platform works seamlessly across various devices and screen sizes.

# CHAPTER- 4

**SYSTEM ARCHITECTURE**

## System Architecture

The **Skill Swap Platform** follows a modular, scalable, and secure architecture that separates concerns between the frontend, backend, and database layers. The system is designed using a **three-tier architecture** comprising:

1. **Presentation Layer (Frontend)**
2. **Application Layer (Backend / Server)**
3. **Data Layer (Database)**
   1. **Overview Diagram**

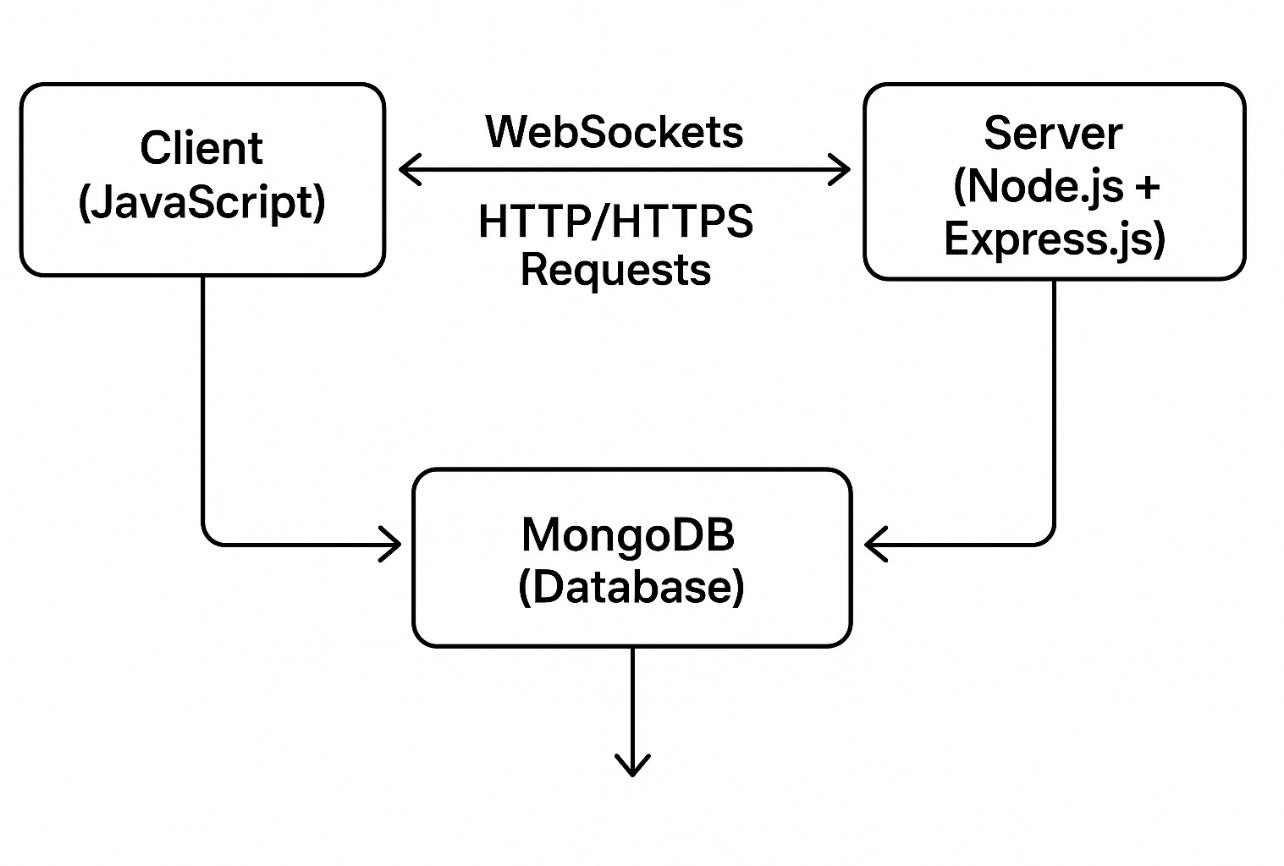


Fig :- 1

* 1. **Components Description**

1. **Presentation Layer (Frontend)**
   * **Technology Used**: CSS, HTML, JavaScript

## Purpose:

oUser interface and user experience

oSkill listing and search functionality

oForms for registration, login, skill addition, and swap requests

oReal-time chat UI and notification popups

* + **Communication**: Uses Axios or Fetch API to interact with backend APIs and Socket.IO for real-time updates.

## Application Layer (Backend)

* + **Technology Used**: Node.js, Express.js

## Purpose:

oHandles all business logic

oAuthenticates users, manages sessions or tokens

oProcesses and stores swap requests, skill data, chat messages

oServes RESTful APIs to the frontend

oSends/receives WebSocket messages for real-time communication

## Data Layer (Database)

* + **Technology Used**: MongoDB (NoSQL)

## Purpose:

oStores users, skills, swap requests, messages, and reviews

oProvides fast, flexible document-based data structure

oEnables indexing for faster search operations

## Real-Time Communication Layer

* + - **Socket.IO** enables:

oReal-time swap request notifications

oLive chat and dashboard updates

oInstant feedback loop between users and server

## External Integration Layer

* + - **Third-Party APIs**:

oWhatsApp and Microsoft Teams API for scheduling and communication

oCloudinary (optional) for file/image uploads

oEmail/Mobile OTP services for user verification

# CHAPTER – 5 MODULE DESCRIPTION

## Module Description

The **Skill Swap Platform** is divided into several core modules, each responsible for a specific set of functionalities. This modular architecture enhances maintainability, reusability, and scalability of the platform.

## User Authentication Module

* + - **Purpose**: Manages user registration, login, logout, and password management.

## Features:

oSecure registration with unique username, email, and mobile number validation.

oPassword encryption using bcrypt.js.

oSession-based or JWT-based authentication.

oPassword reset/change functionality.

* + - **Technologies**: Node.js, Express.js, MongoDB, bcrypt, express-session or JWT

## Profile & Skill Management Module

* + - **Purpose**: Allows users to manage their profiles and add/edit/delete their skills.

## Features:

* + - * User profile with avatar, bio, location, and list of skills.
      * Add new skills with description and proficiency level.
      * Edit or delete existing skills.
    - **Technologies**: React.js (UI), Express.js (API), MongoDB (data storage)

## Skill Discovery and Search Module

* + - **Purpose**: Enables users to explore available skills and find suitable swap partners.

## Features:

* + - * Dynamic search bar and filter options (e.g., category, keyword).
      * Skill listing with user details and ratings.
      * Skill cards with swap option.
    - **Technologies**: React.js, Express.js API endpoints, MongoDB queries

## Swap Request Module

* + - **Purpose**: Facilitates skill swap requests between users.

## Features:

* + - * Send swap request with selected skill.
      * View incoming and outgoing requests.
      * Accept or reject swap requests.
      * Remove swap requests after action is taken.
    - **Technologies**: MongoDB (request tracking), Express.js (logic), Socket.IO (real-time updates)

## Real-Time Communication Module

* + - **Purpose**: Provides live interaction between users once a swap request is accepted.

## Features:

* + - * Real-time chat between users.
      * Notification system for request updates.
      * Dashboard for managing ongoing skill swaps.
    - **Technologies**: Socket.IO, Node.js, React.js

## Scheduling & External Integration Module

* + - **Purpose**: Helps users coordinate swap sessions effectively.

## Features:

* + - * Share calendar or time slot for sessions.
      * Integration with WhatsApp and Microsoft Teams.
      * Resource sharing (e.g., links or tutorials).
    - **Technologies**: External APIs, React.js, Node.js

## Ratings & Reviews Module

* + - **Purpose**: Builds trust in the community by allowing feedback after a swap.

## Features:

* + - * Leave rating (stars) and review after a completed swap.
      * Display user ratings on skill listings and profiles.
    - **Technologies**: MongoDB, Express.js, React.js

# CHAPTER – 6 DATA FLOW DIAGRAM

## Data Flow Diagram (DFD)

The Data Flow Diagram (DFD) helps visualize the flow of data through the system, representing the interaction between users, processes, and data stores. Below are the DFD Levels 0 and 1 to describe the system in detail.

## Level 0 DFD (Context-Level Diagram)

This is the top-level DFD which represents the entire Skill Swap Platform as a single process with its interactions with external entities.

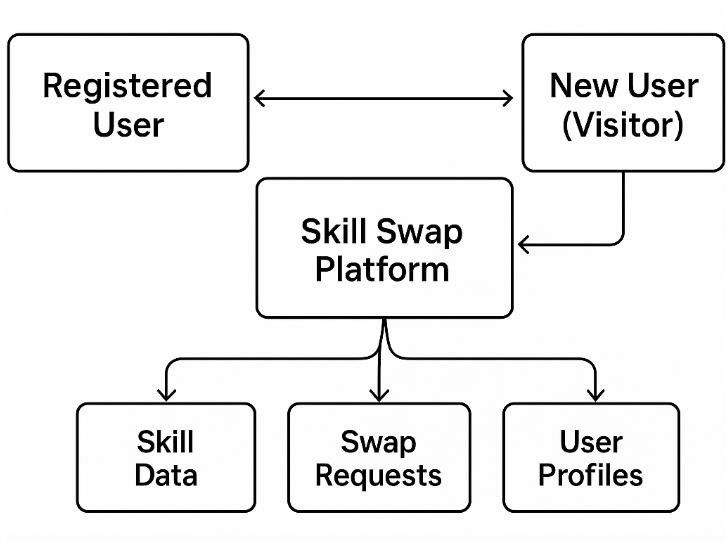


Fig :- 2.1 (Level

## Level 1 DFD

This breaks the main system into its core processes, showing how each one interacts with users and databases.

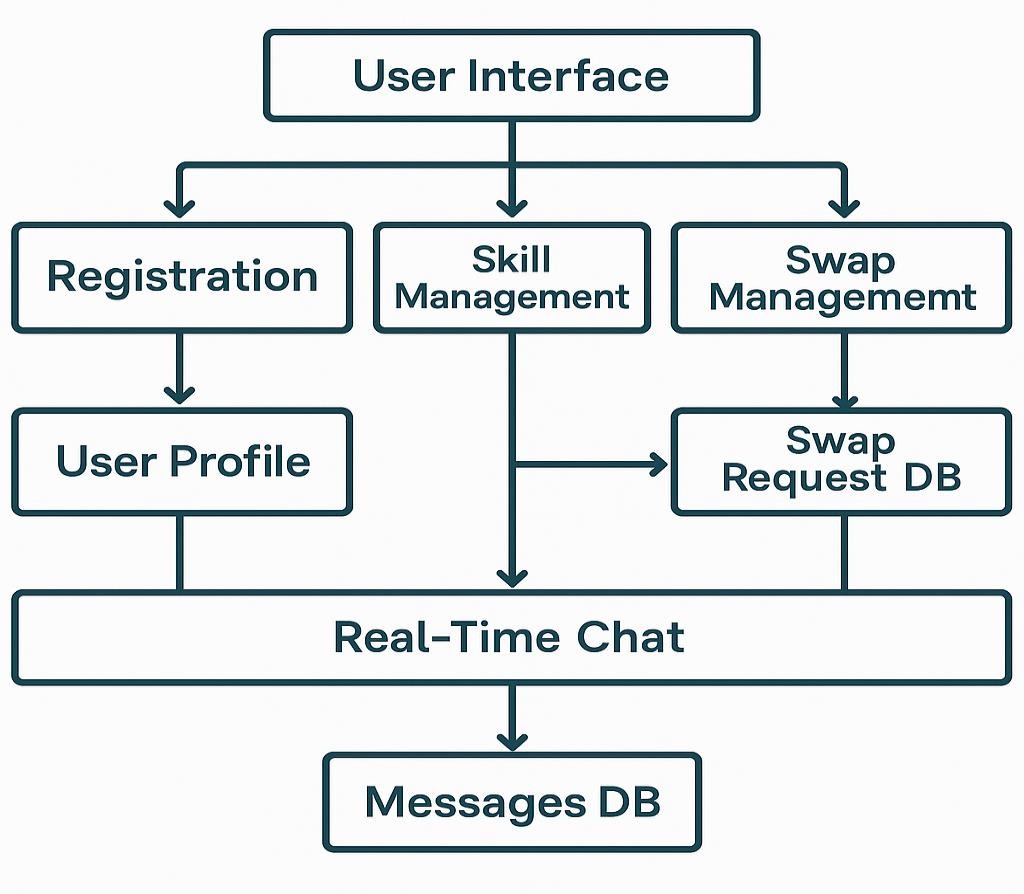


Fig :- 2.2(Level 1)

## Key Processes in Level 1

1. **User Registration & Authentication**
   * Input: User details (name, email, password, etc.)
   * Output: New user record in database, session token

## Skill Management

* + Input: Add/Edit/Delete skill info
  + Output: Updated skill data in skill collection

## Swap Request Management

* + Input: User initiates/accepts/rejects a swap
  + Output: Swap record created/updated/deleted in the database

## Real-Time Communication

* + Input: Messages sent during active swap
  + Output: Stored in messages DB and pushed via WebSockets

1. **Use Case Diagram**

# CHAPTER – 7

**USE CASE DIAGRAM**

The **Use Case Diagram** provides a visual representation of the interactions between the users (actors) and the system (use cases). It helps identify the different functionalities of the platform and how various actors interact with those functionalities.

## Actors

1. **Registered User**: A user who has created an account and logged in.
2. **Visitor**: A user who has not registered or logged in, but can browse the platform.

## Use Case Diagram

Fig.2.3

## Use Cases

1. **Register / Login**:
   * The **Visitor** can register by providing their details (name, email, password, mobile number) and log into the platform. Once registered, they become a **Registered User**.

## Search & View Skills:

* + The **Registered User** can search for skills offered by others, view detailed profiles, and browse the skill database.

## Add / Edit Skill:

* + The **Registered User** can add new skills to their profile, update existing skills, or delete skills if they no longer wish to offer them.

## Request Swap:

* + The **Registered User** can send a swap request to another user for the skills they are interested in.

## Accept / Reject Swap Request:

* + The **Registered User** can accept or reject incoming swap requests. When accepted, the system initiates a discussion and scheduling phase.

## Real-Time Chat:

* + Once the swap request is accepted, both users can chat in real-time to discuss further details, share resources, and coordinate the swap.

## Update Profile / Change Info:

* + The **Registered User** can update their profile information, including their name, bio, and contact details.

## View Profile:

* + The **Registered User** can view their own and others' profiles, including skills, ratings, and reviews.

## Schedule Swap:

* + The **Registered User** can schedule a meeting or session with the other user, integrating external tools like WhatsApp or Microsoft Teams for coordination.

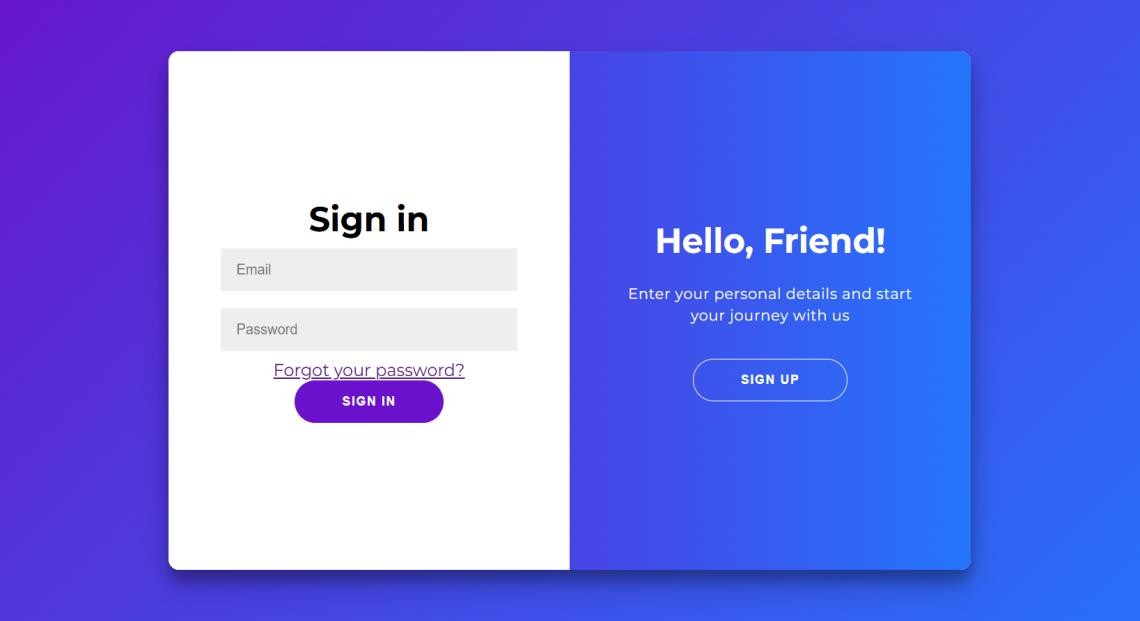
# CHAPTER - 8

**IMPLEMENTATIONS & SCREENSHOTS**

## Home Page

* 1. **Login and Sign up**

Fig.3.1

Fig. 3.2

## Home Page After Login

Fig.3.3

## Profile Page

Fig.3.4

## Skills Section

Fig.3.5

## Swap Request Page

Fig.3.6

## Show Request in Profile

Fig.3.7

## When User Accept The Request then user redirect on whatsapp

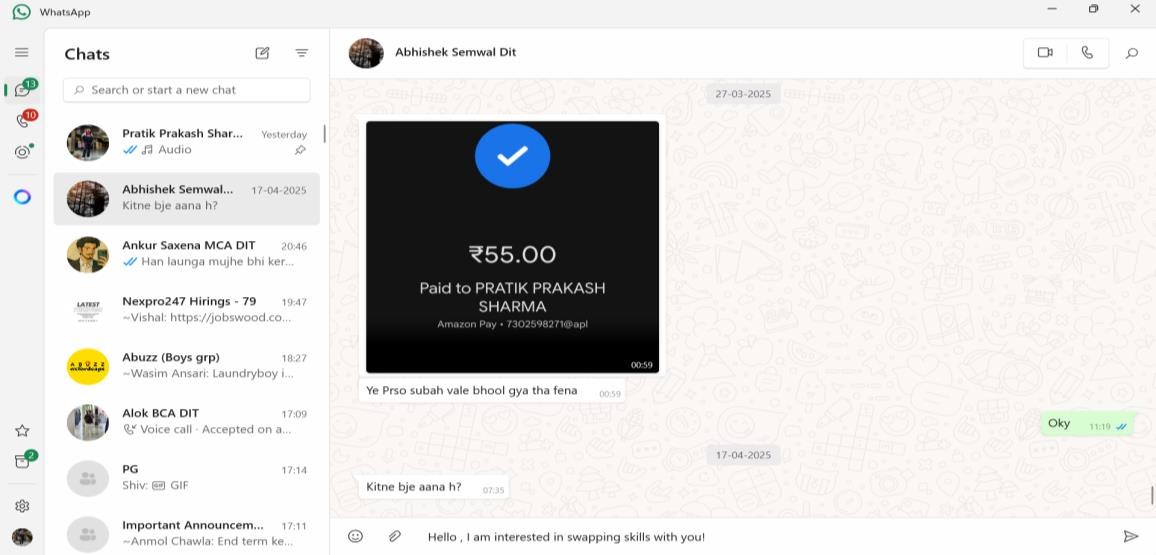


Fig.3.8

1. **Conclusion**

# CHAPTER – 9 CONCLUSION

The Skill Swap Platform is a powerful and innovative web-based application designed to promote collaborative learning and skill exchange among users. By leveraging modern web technologies such as React.js, Node.js, Express, and MongoDB, the platform provides a seamless, user-friendly interface with robust backend support, ensuring a secure and interactive experience.

This platform not only enables users to share and acquire new skills without monetary transactions, but also fosters a sense of community through real-time communication, reviews, and ratings. The integration of real-time features using Socket.IO and scheduling support through tools like WhatsApp or Microsoft Teams ensures practical collaboration beyond just digital interaction.

The project successfully meets its objectives by:

* + Offering a smooth registration and login system.
  + Allowing users to add, view, and manage skills.
  + Enabling swap request handling and real-time notifications.
  + Supporting chat and scheduling for skill exchanges.
  + Promoting trust through user reviews and ratings.

In conclusion, the Skill Swap Platform demonstrates how technology can empower people to share knowledge in a productive and cost-effective way. It opens the door to continuous learning, community building, and personal growth—making it a valuable contribution to the educational and professional development ecosystem.

# CHAPTER- 10

**SCOPE FOR FUTURE WORK**

The **Skill Swap Platform** has been developed with a strong foundation, but there are several opportunities to enhance its features and usability in future versions. These improvements

aim to expand functionality, improve user engagement, and increase system scalability.

## Mobile Application

* + Develop a cross-platform mobile app (using React Native or Flutter) for easier access and a more engaging user experience on mobile devices.

## Video Calling Integration

* + Integrate native video conferencing features directly into the platform instead of relying solely on third-party tools like WhatsApp or Microsoft Teams.

## AI-Based Skill Matching

* + Implement machine learning algorithms to suggest ideal skill swap partners based on user interests, past activity, and rating history.

## Gamification

* + Introduce badges, achievements, and leaderboards to increase motivation and participation among users.

## In-App Notifications

* + Add a notification system to alert users about new requests, accepted swaps, unread messages, and important platform updates.

## Payment Integration (Optional)

* + Allow optional paid sessions or premium mentorship features, integrating secure payment gateways like Razorpay, Stripe, or PayPal.

## Multilingual Support

* + Expand accessibility by adding support for multiple languages, catering to a more diverse and global user base.

## Skill Certification

* + Enable users to receive certification after successful skill swaps, endorsed by both parties, increasing credibility and trust.

## Enhanced Admin Dashboard

* + Provide analytics, user behaviour tracking, and dispute resolution tools for platform administrators.

# CHAPTER- 11 REFERENCES

## References

The following resources, tools, and documentation were referenced and utilized during the development of the **Skill Swap Platform**:

## React.js Documentation

https://reactjs.org/docs/getting-started.html

*Used for building the frontend interface and managing UI components.*

## Node.js Documentation

https://nodejs.org/en/docs

*Used for server-side development and creating a backend runtime environment.*

## Express.js Documentation

<https://expressjs.com/>

*Used for creating RESTful APIs and routing in the backend.*

## MongoDB Documentation

<https://www.mongodb.com/docs/>

*Used as the primary NoSQL database for storing user, skill, and request data.*

## Mongoose Library

<https://mongoosejs.com/>

*Used for modeling MongoDB data and implementing schema-based interaction.*

## Socket.IO Documentation

https://socket.io/docs

*Used for implementing real-time chat and live notifications.*

## Bcrypt.js GitHub Repository

<https://github.com/kelektiv/node.bcrypt.js>

*Used for hashing and securely storing user passwords.*

## Visual Studio Code

<https://code.visualstudio.com/>

*Used as the primary code editor for development.*

## Postman

<https://www.postman.com/>

*Used for API testing and verifying backend routes.*

## MDN Web Docs

<https://developer.mozilla.org/>

*Used for general HTML, CSS, and JavaScript references.*

## GitHub

<https://github.com/>

*Used for version control and project collaboration.*

## WhatsApp API

<https://developers.facebook.com/docs/whatsapp/>

*Referenced for enabling communication through external platforms.*

## Microsoft Teams Integration Docs

<https://learn.microsoft.com/en-us/microsoftteams/platform/>

*Used for exploring scheduling and video session integrations.*

# CHAPTER 1-6

by Pratik Prakash Sharma

**General metrics**

**17,695 2,486 267 9 min 56 sec 19 min 7 sec**

characters words sentences reading

time

speaking time

**Score Writing Issues**

**72**



**91**

Issues left

**14**

Critical

**58**

Advanced

This text scores better than 91% of all texts checked by Grammarly

**Writing Issues**

**Clarity**



**2**

**2** Wordy sentences



**Correctness**



**16**

**1** Misspelled words



**4** Improper formatting



**1** Incorrect phrasing



**1** Comma misuse within clauses



**1** Faulty subject-verb agreement



**5** Determiner use (a/an/the/this, etc.)



**1** Unknown words



**1** Incorrect verb forms



**1** Confused words



**Unique Words 29%**

Measures vocabulary diversity by calculating the percentage of words used only once in your document

unique words

**Rare Words 44%**

Measures depth of vocabulary by identifying words that are not among the 5,000 most common English words.

rare words

**Word Length** **5.8**

Measures average word length characters per word

**Sentence Length** **9.3**

Measures average sentence length words per sentence

# CHAPTER 9-11

by Pratik Prakash Sharma

## General metrics

**4,622 586 63 2 min 20 sec 4 min 30 sec**

characters words sentences reading

time

speaking time

## Score Writing Issues

**10**



**96**

Issues left Critical

**10**

Advanced

This text scores better than 96% of all texts checked by Grammarly

## Unique Words 50%

Measures vocabulary diversity by calculating the percentage of words used only once in your document

unique words

## Rare Words 41%

Measures depth of vocabulary by identifying words that are not among the 5,000 most common English words.

rare words

## Word Length 6.4

Measures average word length characters per word

## Sentence Length 9.3

Measures average sentence length words per sentence