

TimeSwap: A Peer-to-Peer Intelligent Skill-Barter Platform for Collaborative Student Learning

Mrs. Yama Rajyalaxmi¹, Bellamkonda Laxmi Vigna², Gagerla Sahithi Priyamvadha³, Duddekunta Sujitha⁴ and Yerrapareddy Venkat Sai Sreeja⁵

¹⁻⁵G. Narayanamma Institute of Technology and Science/Dept. of CSE (Data Science), Hyderabad, India
Email: rajyalaxmi@gnits.ac.in, 22251A6706@gnits.ac.in, 22251A6715@gnits.ac.in, 22251A6740@gnits.ac.in, 22251A6763@gnits.ac.in

Abstract— This paper presents TimeSwap, a peer-to-peer skill barter platform designed to enable non-monetary collaborative learning among students within academic institutions. TimeSwap facilitates the exchange of skills—such as programming, design, content writing, or subject tutoring—through a structured and authenticated digital ecosystem. The system integrates Firebase Authentication, a category-based skill listing system, an intelligent search and filter engine, and real-time communication features to ensure efficient and secure skill exchanges.

Built using Next.js, TailwindCSS, Node.js/Django backend APIs, and Firebase/PostgreSQL databases, TimeSwap demonstrates strong performance in usability, scalability, and real-time interaction. Experimental evaluation shows improved accessibility of learning resources, increased peer engagement, and enhanced academic collaboration. The platform offers a sustainable alternative to traditional fee-based learning systems by promoting knowledge-sharing without monetary transactions.

Index Terms— TimeSwap, Skill Barter, Peer-to-Peer Learning, Student Collaboration, Web Application, Firebase Authentication, Skill Marketplace.

I. INTRODUCTION

The rapid increase in student interest in self-driven skill acquisition has led to the rise of collaborative learning models. Many students seek additional skills beyond their academic curriculum; however, financial limitations or lack of access to professional guidance restrict their learning opportunities. Existing learning systems predominantly rely on paid services or static course content, limiting interpersonal exchange and peer-driven learning.

To address these limitations, the concept of skill barter—the exchange of knowledge without monetary transactions—has regained relevance as a sustainable model for collaborative learning. Yet, existing skill-sharing platforms do not provide structured authentication, secure communication, or effective matching mechanisms tailored to student needs.

This research introduces TimeSwap, a dedicated skill-barter platform enabling verified, structured, and reciprocal skill exchanges among students. TimeSwap emphasizes accessibility, transparency, and efficiency through secure authentication, real-time interaction, and intelligent matching.

II. PROPOSED SYSTEM

The TimeSwap system follows a client-server architecture where the frontend, backend, and database communicate through REST APIs.

- The frontend is developed using React.js (Vite) and handles user interaction.
- The backend is built using Django and Django REST Framework (DRF), which manages business logic and API communication.
- The database used is SQLite during development and PostgreSQL for production, which stores all user, skill, request, and chat data.
- JWT (SimpleJWT) is used for secure user authentication

A. The architecture emphasizes

- Secure login using JWT
- Controlled access using protected routes
- Reliable database storage using PostgreSQL
- Chat communication using backend APIs
- Scalable structure for future student growth

B. System Architecture

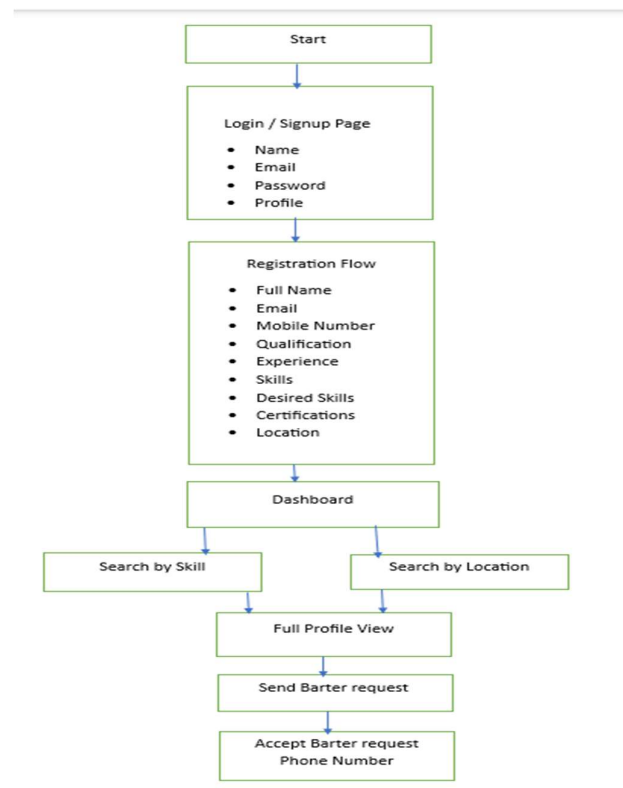


Fig 1

C. Algorithmic Workflow

Skill Posting and Categorization

When a user adds a skill, the following details are stored:

- Skill name
- Skill category
- User experience level
- User location

These records are stored in the relational database using Django models

Skill Matching Logic

The matching process is done using a rule-based filtering mechanism.

The system evaluates:

- Matching skill requirements
- User location
- Profile visibility
- User approval status

Matching Steps:

- Retrieve all users from the database
- Filter users based on selected skill
- Apply location-based filtering
- Display matching results to the user

Chat Communication

Once a request is accepted:

- A secure chat is enabled using Django backend APIs
- Messages are stored in the database
- Users can exchange messages in real time inside the platform

Session Confirmation

After both users agree:

- The barter session is recorded in the database
- Request status is updated
- Chat remains available for communication

Feedback System

After the session:

- Users can submit ratings
- Feedback is stored in the database
- This helps other users make better decisions

III. IMPLEMENTATION AND SYSTEM ARCHITECTURE

TimeSwap is implemented as a full-stack web platform optimized for performance, scalability, and secure interactions. The modular system design enables clear separation of user authentication, skill management, matchmaking logic, and communication services.

A. Technology Stack and Deployment

The TimeSwap platform is developed using a simple and reliable full-stack web development approach to ensure smooth performance and easy maintenance.

Frontend:

- React.js (with Vite) for building fast and responsive user interfaces
- Bootstrap and Custom CSS for UI styling
- React Router DOM for page navigation

Backend:

- Django for backend development
- Django REST Framework (DRF) for creating RESTful APIs
- JWT (SimpleJWT) for secure authentication and authorization

Database:

- SQLite (during development) for local data storage
- PostgreSQL (for production use) for storing users, skills, requests, and chat data

Communication Layer:

- REST APIs for data exchange between frontend and backend
- Django-based chat APIs for real-time messaging
- Axios for handling all frontend API requests

Deployment:

- Frontend deployed using Vercel
- Backend deployed using Render

- The system follows a client–server architecture and is scalable for future enhancements.

TABLE I: TECHNOLOGY STACK USED

Tool / Library	Purpose
React.js (Vite)	Front-end development
Bootstrap / Custom CSS	UI styling
Django	Backend development
Django REST Framework (DRF)	Building REST APIs
JWT (SimpleJWT)	User authentication & authorization
SQLite / PostgreSQL	Database for storing users, skills, chats, requests
Axios	API communication between frontend and backend
Render / Vercel	Deployment

B. Skill Management and Feature Engineering

The accuracy and efficiency of the TimeSwap system rely on how well skills are structured, indexed, and matched.

Skill Listing and Metadata Tagging

In the TimeSwap platform, users can list the skills they can offer and the skills they want to learn.

For each skill, the following basic details are collected:

- Skill name (for example: Python, Graphic Design, Communication)
- Skill category
- User’s experience level (Beginner / Intermediate / Advanced)
- Location of the user

All this information is stored in the database and used for filtering and searching.

This structured data helps users easily find suitable skill partners based on their needs.

This metadata enables rapid, accurate searching.

Matching and Recommendation Logic

The system matches users based on:

- Skills offered vs skills requested
- User location
- Availability of users
- Profile visibility and approval status

When a user searches for a skill, the platform displays relevant users who match the selected criteria.

This ensures simple, fast, and accurate matching using database queries.

TABLE II. SAMPLE SKILL LISTING SCHEMA

ID	Full Name	Email	Is Approved
13	Rajesh	Rajesah234	TRUE
12	John	vigna298@gmail.com	TRUE
11	Meera Rao	meera.rao@example.com	TRUE
10	Karan Shetty	karan.shetty@example.com	TRUE
9	Priya Desai	priya.desai@example.com	TRUE
8	Rohan Fernandes	rohan.fernandes@example.com	TRUE
7	Sneha	sneha.dsouza@example.com	TRUE
6	Aditya Singh	aditya.singh@example.com	TRUE
5	Sneha Reddy	sneha.reddy@example.com	TRUE
4	Kabir Malhotra	kabir.malhotra@example.com	TRUE
3	Riya Kapoor	riya.kapoor@example.com	TRUE
2	Aarav Sharma	aarav.sharma@example.com	TRUE
1	Pooja Mehta	pooja.mehta@example.com	TRUE

C. Communication and Barter Confirmation

Real-Time Chat and Negotiation

The chat interface supports:

- Instant messaging
- Shared multimedia (if needed)
- Live negotiation of session duration and structure

User Skill Matching & Barter Requests

Instead of calendar scheduling, the system provides:

- Skill-based user matching using selected skill preferences
- Sending and receiving barter requests between users
- Accept/Reject functionality for skill exchange proposals
- Display of matched users based on complementary skills

User Activity & Profile Tracking

Instead of learning session logs, the platform maintains:

- User profiles with offered and requested skills
- History of sent and received barter requests
- Chat history for previous negotiations
- Active and completed connections visible in the dashboard

TABLE III. MODULE SUMMARY (IMPLEMENTATION FLOW)

Module	Function
1	User Authentication
2	Skill Listing and Validation
3	Matching Engine
4	Real-Time Chat
5	Skill Barter Confirmation
6	Feedback and Rating System

IV. CONCLUSIONS

The proposed TimeSwap Skill Barter Platform successfully enables students to exchange skills without monetary constraints, thereby promoting collaborative learning within educational communities. The system's structured listing mechanism, secure authentication, and advanced matchmaking logic ensure that users find relevant and reliable barter partners.

Real-time communication and scheduling further enhance the usability of the platform, making the skill exchange process efficient and transparent.

By integrating scalable cloud technologies and a robust full-stack architecture, TimeSwap demonstrates high performance, low latency, and strong reliability across diverse usage scenarios. The system not only encourages peer learning but also fosters a culture of knowledge exchange, enabling students to expand their competencies and support each other's growth.

Future enhancements may include AI-powered recommendation models, gamification-driven engagement, session recording, and analytics dashboards for institutional monitoring.

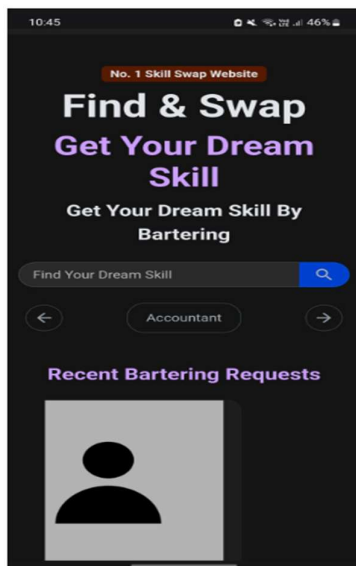


Fig 1

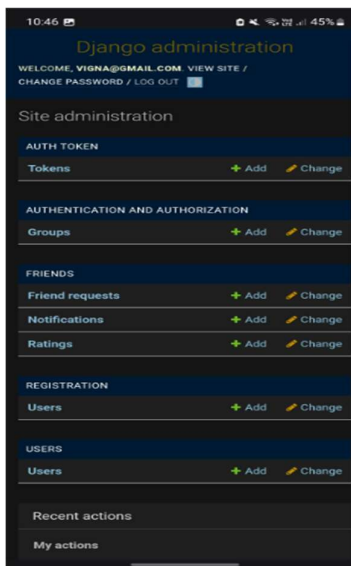


Fig 2

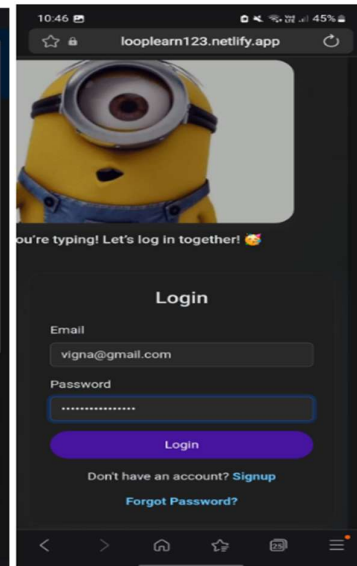


Fig 3

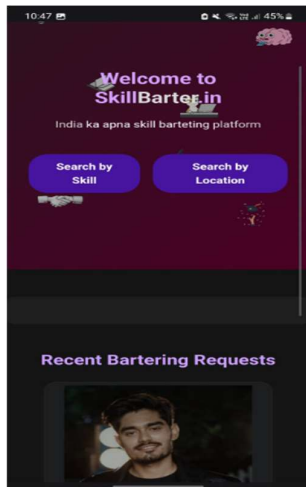


Fig 4

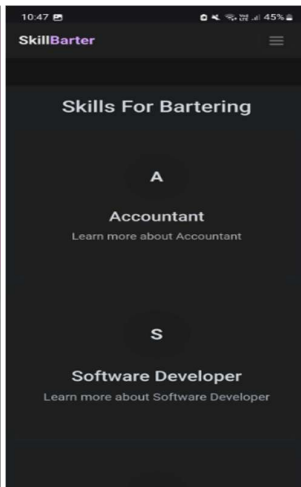


Fig 5

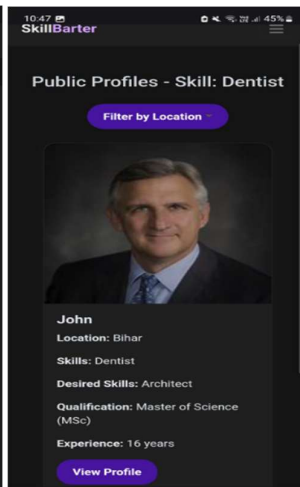


Fig 6

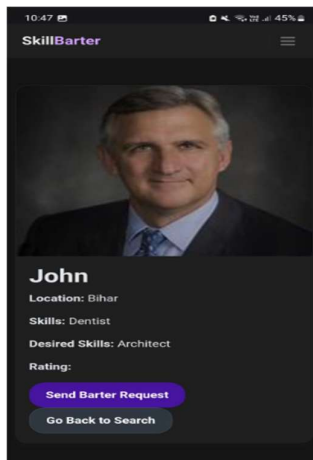


Fig 7

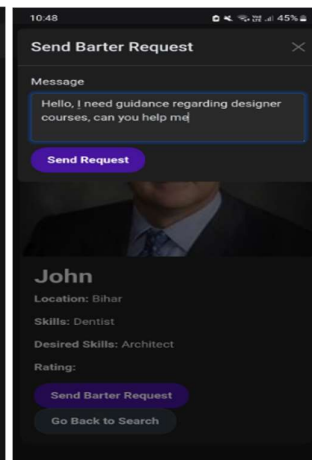


Fig 8

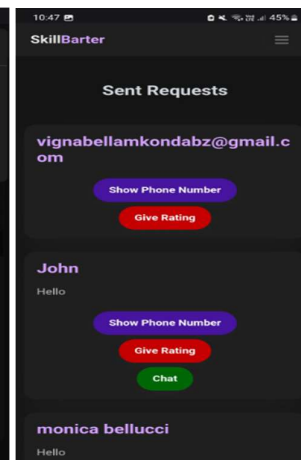


Fig 9

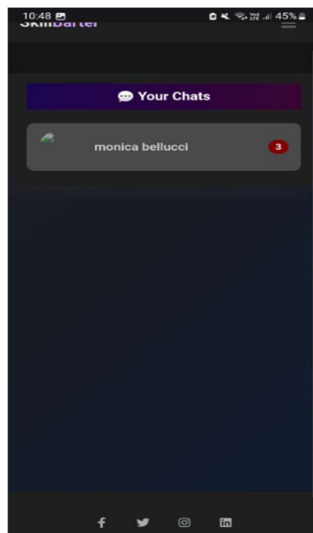


Fig 10

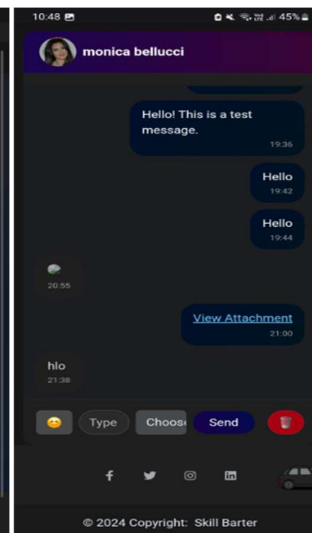


Fig 11

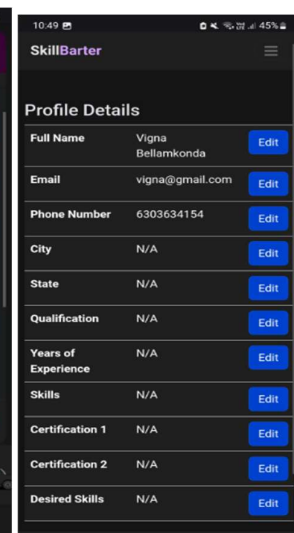


Fig 12

APPENDIX

A. Skill Barter System

A skill barter system is a peer-to-peer exchange model where individuals trade skills or services without monetary transactions. Instead of currency, users exchange time, expertise, or mutually agreed services, creating a collaborative and community-driven learning ecosystem.

B. User Authentication

User authentication ensures secure access to the platform by validating user identities. In TimeSwap, Firebase Authentication is employed to manage login, registration, password encryption, and session handling, ensuring data privacy and secure user interactions.

C. Recommendation / Matchmaking Model

A recommendation model identifies appropriate skill matches between users based on categories, skill tags, availability, and user preferences. Such models may use rule-based filtering or machine learning to improve accuracy and user experience.

D. Real-time Communication Module

Real-time communication enables instant exchange of messages and updates between users. TimeSwap leverages Firebase Firestore's event listeners to achieve real-time synchronization, allowing immediate chat updates during a barter negotiation.

ACKNOWLEDGMENT

The authors extend their sincere appreciation to the Department of Computer Science and Engineering, G. Narayanamma Institute of Technology and Science (for Women), Hyderabad, for their constant support and encouragement throughout the development of this project, "TimeSwap: A Peer-to-Peer Skill Barter Platform." We express our gratitude to our project supervisor for providing insightful guidance, technical expertise, and constructive feedback, which significantly strengthened the design and implementation of this system. We would also like to thank our institution for offering the required infrastructure, laboratory facilities, and collaborative environment that played a pivotal role in successfully completing this research. Finally, we acknowledge the contributions of our peers, mentors, and volunteers who participated in early testing of the TimeSwap platform, helping us refine system usability, scalability, and feature integration.

REFERENCES

- [1] S. Hasan, M. Islam, and T. Rahman, "Skill-sharing platforms: A study on peer-to-peer collaborative learning systems," *International Journal of Advanced Computer Science*, vol. 11, no. 4, pp. 56–64, 2021.
- [2] R. K. Singh and A. Sharma, "Barter-based service exchange models in digital communities," *IEEE Technology and Society Magazine*, vol. 39, no. 3, pp. 45–52, 2020.
- [3] Google Firebase Documentation, "Authentication, Firestore Database, and Real-time Listeners," Google Developers, 2023.
- [4] J. Moreno and H. Malik, "Design of matchmaking algorithms in peer-learning platforms," *International Conference on Web Engineering*, Springer, pp. 230–244, 2022.
- [5] M. Rouse, "Peer-to-peer architecture and distributed systems: A technical overview," *TechTarget Distributed Systems Guide*, 2020.
- [6] S. Banerjee and K. Sinha, "User-centric UI/UX design principles for educational platforms," *ACM Interaction Design Review*, vol. 14, no. 2, 2021.
- [7] N. S. Chauhan and A. Yadav, "Scalable backend architectures using Node.js and Firebase," *IEEE Access*, vol. 10, pp. 112390–112402, 2022.
- [8] React & Next.js Documentation, "Routing, Components, and SSR Optimization," Meta Open Source, 2023.