SkillSwap

Real-Time Resource Pool (RTRP)

A Web-Based Peer-to-Peer Skill Exchange Platform

Course: [Insert Course Name]

Guide/Mentor: [Insert Mentor Name]

Team Members & Roles:

[Member 1 Name] Frontend Development (React UI)

February 1, 2026

Contents

1 Introduction **3**
1.1 Key Concept . . . 3

2 Problem Statement & Motivation **3**
2.1 Key Problems Identified . . . 3
2.2 Impact Assessment . . . 3

3 Objectives & Goals **3**

4 Literature Review / Background **4**
4.1 Existing Solutions Analysis . . . 4
4.2 Gap Analysis . . . 4

5 Methodology **4**
5.1 Development Approach . . . 4
5.2 Technology Stack . . . 5

6 System Design & Architecture **5**
6.1 Frontend Architecture . . . 5
6.1.1 Core Components . . . 5
6.1.2 Page Components . . . 5
6.1.3 Reusable Components . . . 5
6.2 Backend Architecture . . . 6
6.2.1 Middleware Layer . . . 6
6.2.2 API Endpoints . . . 6
6.3 System Flow Diagrams . . . 6
6.4 Authentication Flow . . . 7
6.4.1 Signup Process . . . 7
6.4.2 Login Process . . . 8
6.4.3 Security Features . . . 8
6.5 Database Schema . . . 8
6.5.1 Users Table . . . 8
6.5.2 Skills Table . . . 9
6.5.3 Messages Table . . . 9
6.5.4 Database Relationships . . . 9
6.5.5 Entity Relationship Diagram . . . 10
6.5.6 Database Constraints . . . 10

7 Implementation Details **10**
7.1 Authentication System . . . 10
7.2 Skill Marketplace . . . 10
7.3 Real-Time Chat System . . . 11
7.4 User Profiles . . . 11
7.5 Dashboard . . . 11

8 Application Screenshots **11**
8.1 Dashboard . . . 11

8.2 Browse Skills Marketplace 12
8.3 Browse Community Members 12
8.4 My Skills Management 13
8.5 User Profile 13
8.6 Messaging System 14

9 Testing & Database Results **14**
9.1 Testing Strategy 14
9.2 Test Results Summary 15
9.3 Implementation Status 15
9.4 Database Test Results 15
9.4.1 Platform Statistics Overview 15
9.4.2 Registered Users Data 16
9.4.3 Skills Marketplace Data 16
9.4.4 Chat Messages Data 17
9.4.5 Data Verification Summary 17

10 Challenges & Solutions **17**
10.1 Personal Learnings 17
10.2 Technical Challenges & Solutions 18

11 Future Scope **18**

12 Conclusion **18**

References **18**

Acknowledgments **19**

1 Introduction

SkillSwap is a Real-Time Resource Pool (RTRP) — a web-based platform designed to facilitate peer-to-peer skill exchange without the need for money. It operates on a "Time Banking" concept where **time is the currency**.

In a world where specialized education is often expensive, SkillSwap aims to **democratize learning** by allowing anyone to teach what they know and learn what they don't, simply by trading their time.

1.1 Key Concept

- **Time-Based Economy:** 1 hour of teaching = 1 hour of learning credit
- **Peer-to-Peer:** Direct connection between learners and teachers
- **Community-Driven:** No monetary transactions involved

2 Problem Statement & Motivation

2.1 Key Problems Identified

1. **High Cost of Education:** Professional tutoring and courses are often expensive and inaccessible to many.
2. **One-Way Learning:** Traditional platforms are often just consumption-based (videos), lacking interactive feedback and personal connection.
3. **Undervalued Skills:** Many people have valuable skills (cooking, coding, music) but no easy platform to share them for return value.
4. **Lack of Community:** It's hard to find a local or dedicated learning partner for mutual growth.

2.2 Impact Assessment

This affects **students, hobbyists, and professionals** who want to upskill but are limited by budget constraints or lack of connections to knowledgeable teachers.

3 Objectives & Goals

The primary objectives of the SkillSwap project are:

1. **Create a Time-Based Economy:** Develop a system where 1 hour of teaching equals 1 hour of learning credit.
2. **Facilitate Peer-to-Peer Connection:** Enable users to find others based on specific skills they want to learn or teach.
3. **Ensure Real-Time Interaction:** Provide chat and video capabilities for seamless communication between users.

4. **User-Friendly Interface:** Build a responsive, modern web application for easy navigation across all devices.

4 Literature Review / Background

4.1 Existing Solutions Analysis

Table 1: Comparison of Existing Platforms

Platform	Strengths	Limitations
Coursera/Udemy	Great content, structured courses	One-way, expensive, no peer interaction
Upwork/Fiverr	Professional marketplace	Money-focused, transactional
Traditional Time Banks	Community-based exchange	Offline, localized, lacks modern tools

4.2 Gap Analysis

Existing solutions either **cost money** or lack the structured "skill-for-skill" direct exchange mechanism. SkillSwap bridges this gap by modernizing the time-bank concept with a dedicated web application featuring:

- Real-time communication tools
- Digital skill marketplace
- Time credit tracking system
- Modern, responsive user interface

5 Methodology

5.1 Development Approach

Agile Development: The project was developed in iterative sprints:

1. Sprint 1: Backend API development
2. Sprint 2: Frontend UI development
3. Sprint 3: Integration and testing
4. Sprint 4: Polish and deployment

Iterative Testing: Each feature (Authentication, Chat, Skills) was tested individually before integration.

5.2 Technology Stack

Table 2: Technology Stack Used

Layer	Technology
Frontend	React 19 + Vite + Tailwind CSS
Backend	Node.js + Express.js
Database	MySQL (with mysql2 driver)
Authentication	JWT (JSON Web Tokens) + bcrypt
State Management	React Context API

6 System Design & Architecture

6.1 Frontend Architecture

The frontend is built as a **Single Page Application (SPA)** using React 19 with the following structure:

6.1.1 Core Components

- **React Router DOM:** Handles client-side navigation
- **AppContext Provider:** Manages global state (user data, skills, chats)
- **Vite:** Fast development server and build tool
- **Tailwind CSS:** Utility-first responsive styling

6.1.2 Page Components

- Login & Signup Pages
- Dashboard
- Browse Skills
- My Skills
- Profile
- Messages

6.1.3 Reusable Components

- Sidebar navigation
- SkillCard display
- ChatPanel for messaging
- VideoCall interface
- Toast notifications

6.2 Backend Architecture

The backend follows a **RESTful API** architecture with Express.js:

6.2.1 Middleware Layer

- **CORS:** Cross-Origin Resource Sharing
- **JSON Parser:** Request body parsing
- **JWT Auth:** Token verification middleware

6.2.2 API Endpoints

Table 3: REST API Endpoints

Route	Method	Description
/api/auth/signup	POST	Register new user
/api/auth/login	POST	Login & get JWT token
/api/user	GET	Get current user profile
/api/user/all	GET	Get all users (public)
/api/user/logout	POST	Set user offline
/api/skills	GET/POST	Get all skills / Create skill
/api/skills/:id	GET/DELETE	Get/Delete specific skill
/api/chats	GET	Get all user conversations
/api/chats/:userId	GET/POST	Get/Send messages

6.3 System Flow Diagrams

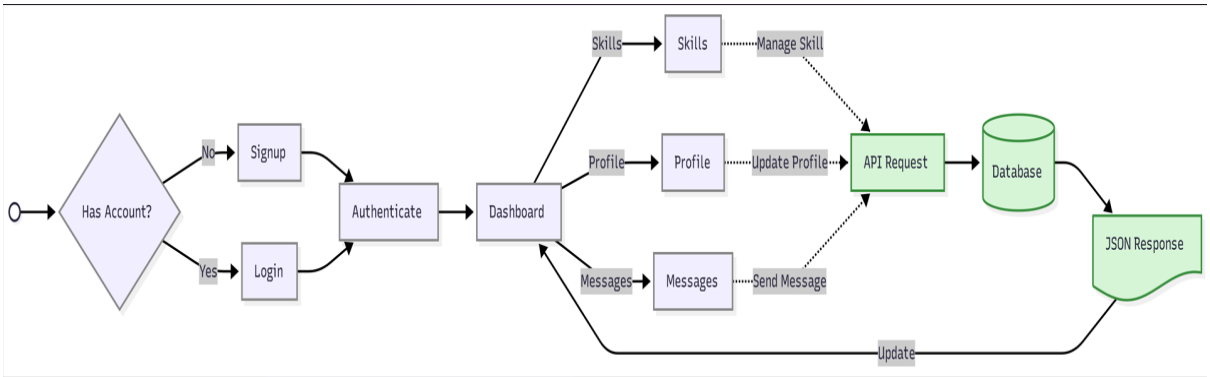

Figure 1: Complete System Flow – User journey from authentication to skill exchange

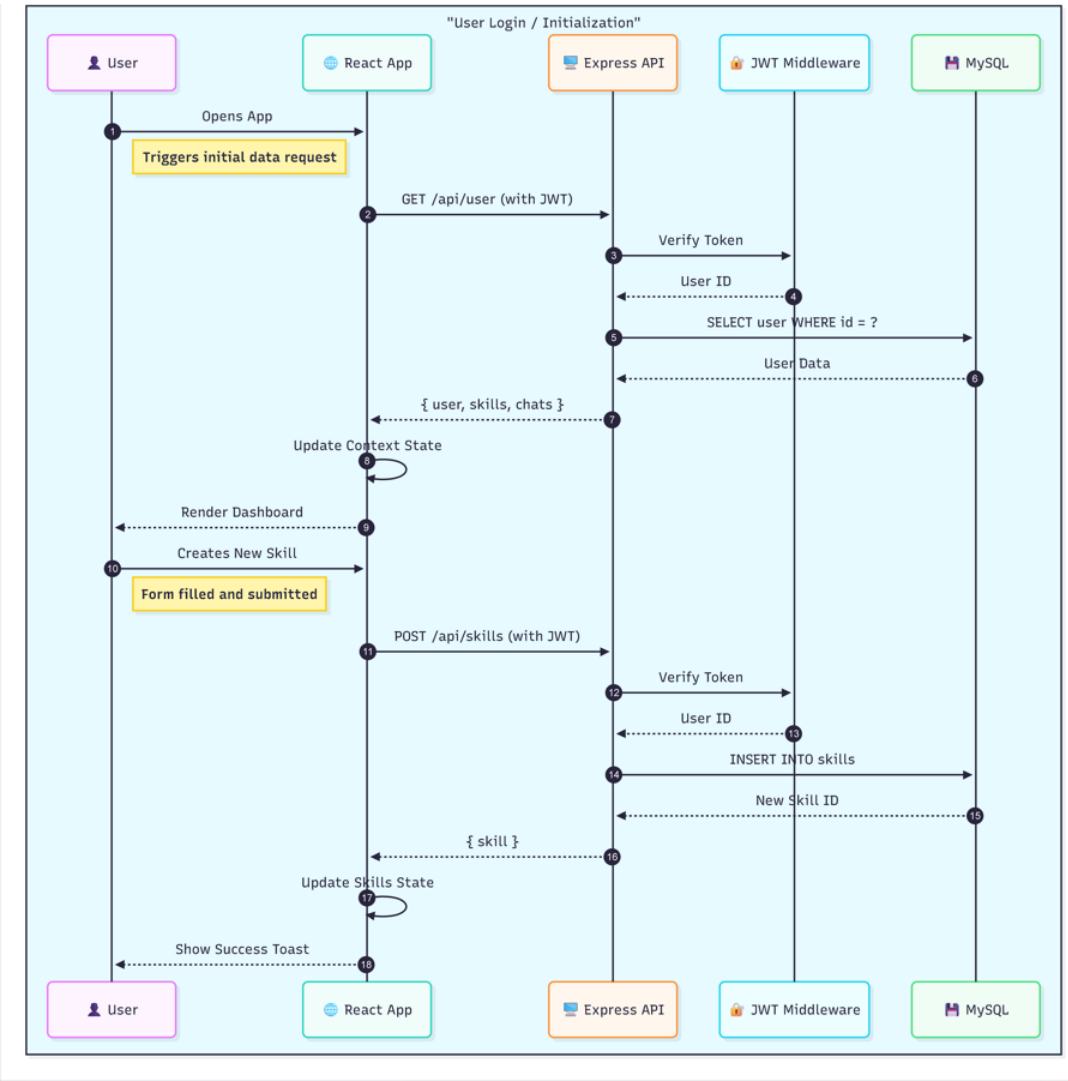

Figure 2: Complete Data Flow – Request-response cycle between frontend, API, and database

6.4 Authentication Flow

6.4.1 Signup Process

1. User fills signup form (username, email, password)
2. Frontend sends POST request to /api/auth/signup
3. Backend hashes password with bcrypt (10 salt rounds)
4. User record inserted into MySQL database
5. JWT token generated and returned
6. Token stored in localStorage
7. User redirected to Dashboard

6.4.2 Login Process

1. User enters email and password
2. Frontend sends POST request to /api/auth/login
3. Backend retrieves user record from database
4. Password compared using bcrypt
5. User's online status updated to TRUE
6. JWT token generated and returned
7. AppContext updated with user data
8. User redirected to Dashboard

6.4.3 Security Features

✓ Password hashing with bcrypt (10 salt rounds)

✓ JWT tokens with 1-hour expiration

✓ Protected routes with auth middleware

✓ Real-time online status tracking

6.5 Database Schema

6.5.1 Users Table

Table 4: Users Table Structure

Column	**Type**	**Description**
id	INT (PK)	Auto Increment
username	VARCHAR	Unique Username
email	VARCHAR	Unique Email
password	VARCHAR	Hashed with bcrypt
avatar	TEXT	Profile Image URL
timeCredits	INT	Default: 10
bio	TEXT	User Biography
role	VARCHAR	Job Title
location	VARCHAR	City/Country
website	VARCHAR	Personal URL
is_online	BOOLEAN	Online Status
last_seen	TIMESTAMP	Last Activity
created_at	TIMESTAMP	Registration Date

6.5.2 Skills Table

Table 5: Skills Table Structure

Column	Type	Description
id	INT (PK)	Auto Increment
title	VARCHAR	Skill Name
description	TEXT	Detailed Description
category	VARCHAR	Programming/Music/etc
hours	DECIMAL	Time Credit Value
user_id	INT (FK)	References Users
created_at	TIMESTAMP	Posted Date

6.5.3 Messages Table

Table 6: Messages Table Structure

Column	Type	Description
id	INT (PK)	Auto Increment
sender_id	INT (FK)	Who Sent
receiver_id	INT (FK)	Who Receives
text	TEXT	Message Content
created_at	TIMESTAMP	Sent Time

6.5.4 Database Relationships

Table 7: Entity Relationships

Relationship	Type	Description
Users → Skills	One-to-Many	One user can create many skills
Users → Messages (sender)	One-to-Many	One user can send many messages
Users → Messages (receiver)	One-to-Many	One user can receive many messages

6.5.5 Entity Relationship Diagram

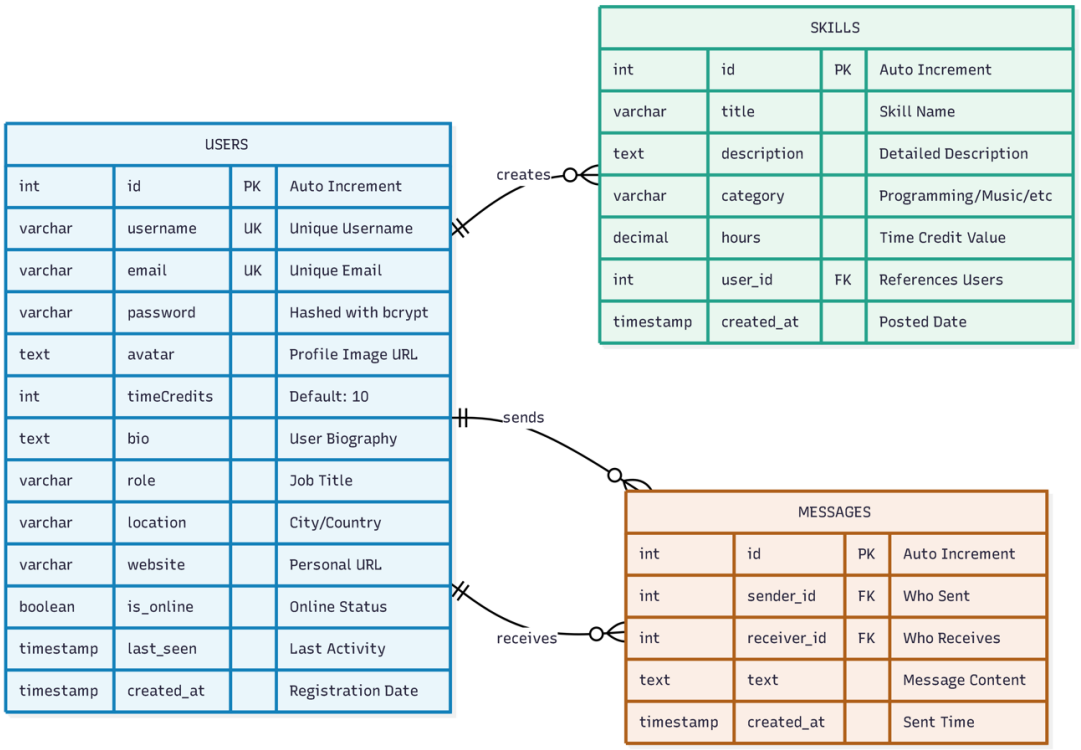

Figure 3: Database Architecture – Entity relationships and table structure

6.5.6 Database Constraints

- `ON DELETE CASCADE` for skills (delete user = delete their skills)
- `ON DELETE CASCADE` for messages (delete user = delete their messages)
- `UNIQUE` constraints on username and email

7 Implementation Details

7.1 Authentication System

- Secure user registration and login
- JWT token-based authentication
- Password hashing with bcrypt
- Session persistence with localStorage

7.2 Skill Marketplace

- Create skills with Title, Description, Category, Hours
- Browse all available skills from community
- Filter skills by category (Programming, Music, Design, etc.)

- Real-time online status indicator for skill owners

7.3 Real-Time Chat System

- Persistent chat messages stored in MySQL
- Message history per conversation
- Real-time message sending and receiving

7.4 User Profiles

- Customizable avatars (DiceBear integration)
- Bio, role, location, and website fields
- View your own skills on profile page
- Time credits balance display

7.5 Dashboard

- View community statistics (members, skills, online users)
- Quick access to browse marketplace
- Featured skills grid
- Member discovery section

8 Application Screenshots

This section showcases the key user interfaces of the SkillSwap application.

8.1 Dashboard

The main dashboard provides an overview of the platform, showing community statistics, featured skills, and quick navigation options.

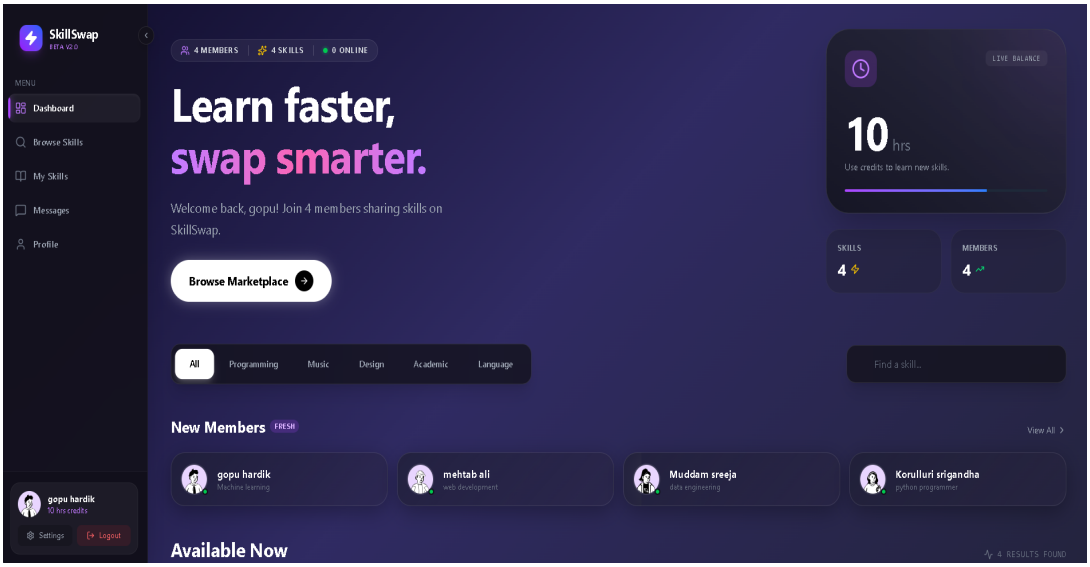

Figure 4: Dashboard – Community overview with statistics and featured skills

8.2 Browse Skills Marketplace

Users can browse all available skills posted by community members, with filtering options by category.

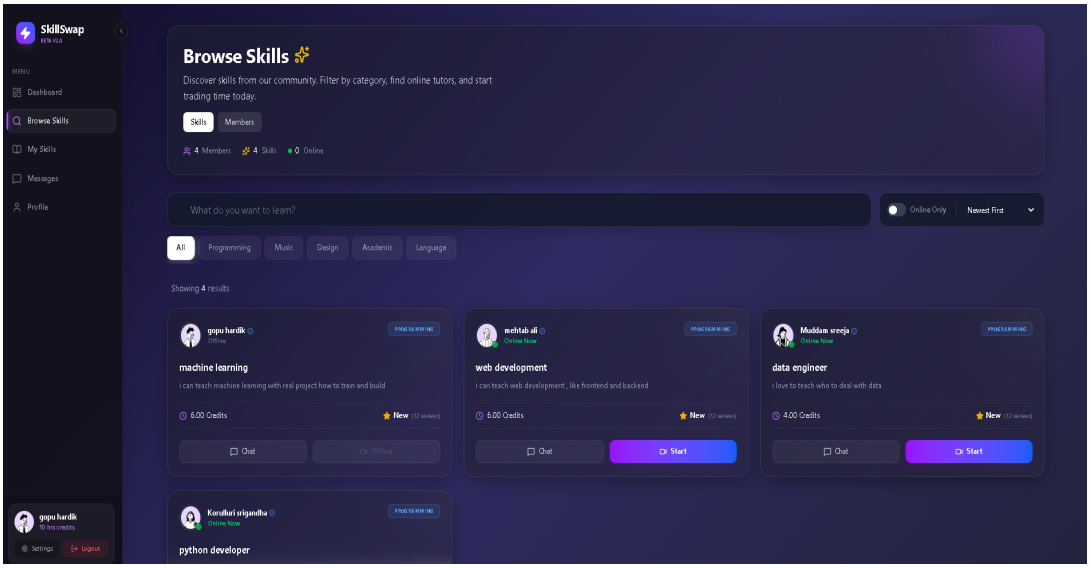

Figure 5: Browse Skills – Skill marketplace with category filters

8.3 Browse Community Members

The members page displays all registered users with their expertise areas and online status indicators.

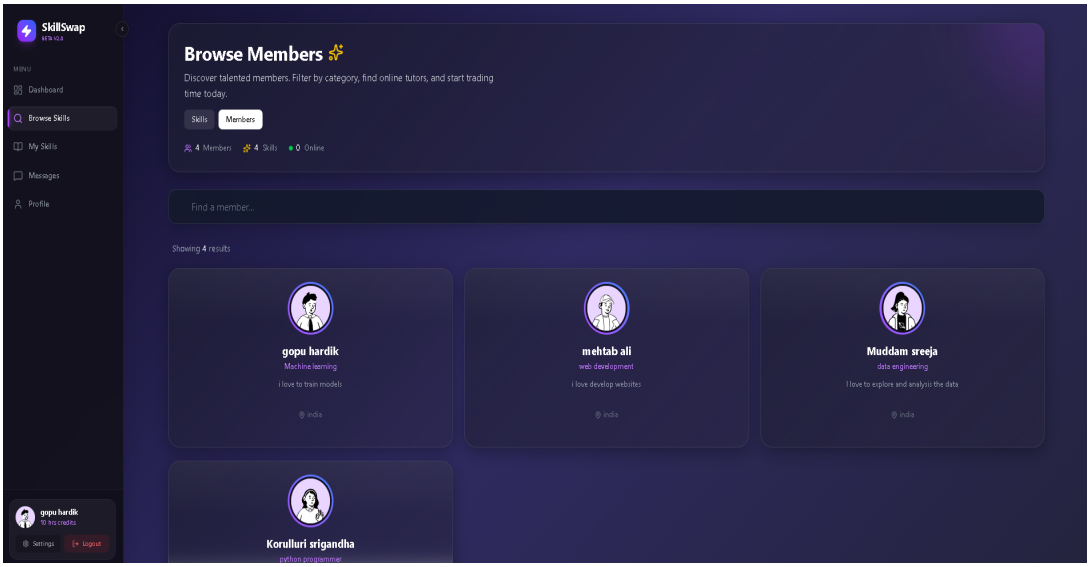

Figure 6: Browse Members – Discover community members and their expertise

8.4 My Skills Management

Users can manage their own skills – create new offerings, edit existing ones, or remove skills they no longer offer.

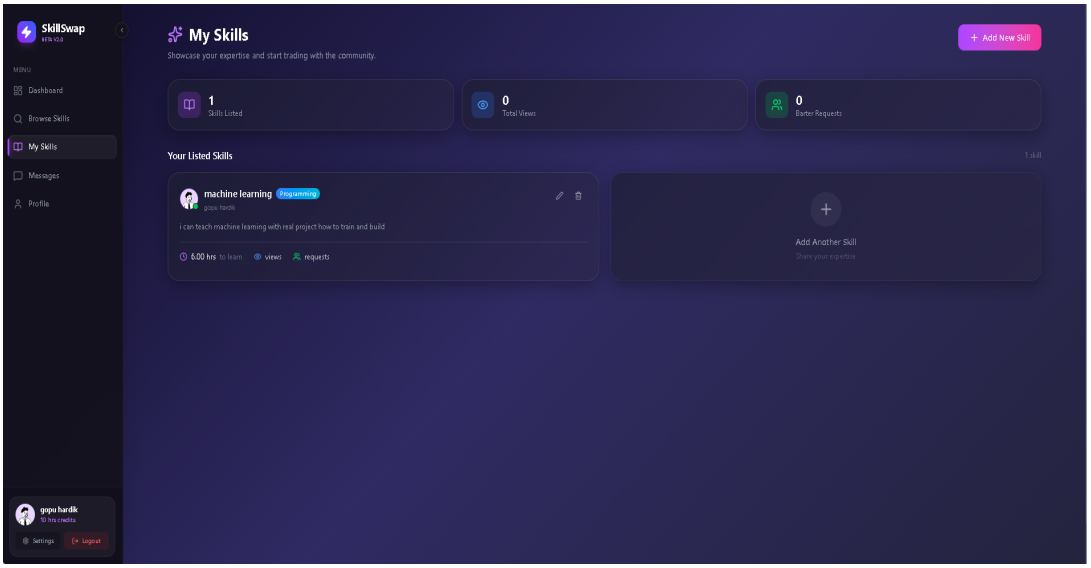

Figure 7: My Skills – Personal skill portfolio management

8.5 User Profile

The profile page displays comprehensive user information including bio, role, location, and listed skills.

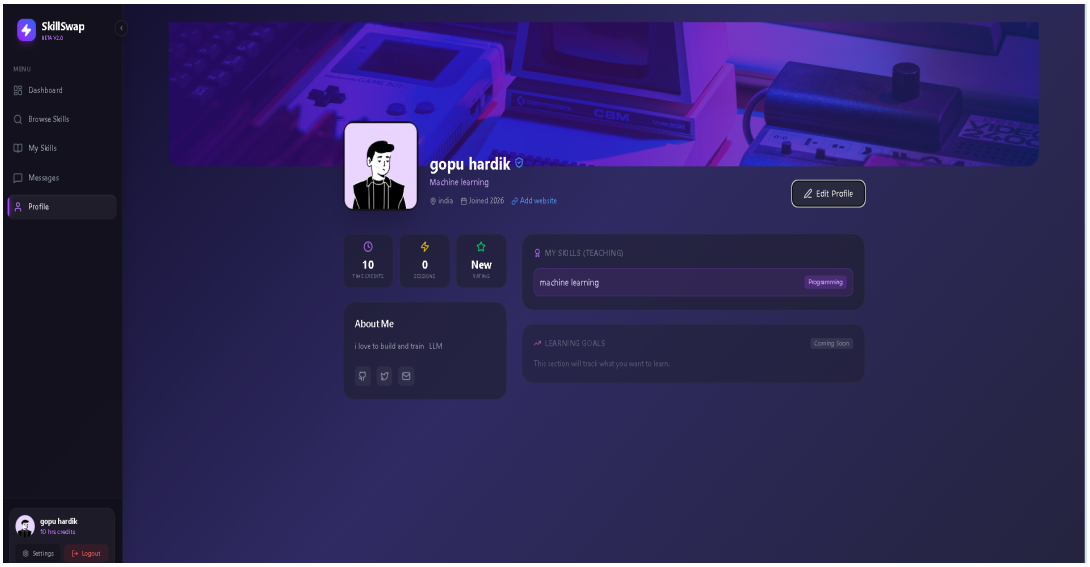

Figure 8: User Profile – Personal information and skill portfolio display

8.6 Messaging System

The chat interface enables real-time communication between users for skill exchange coordination.

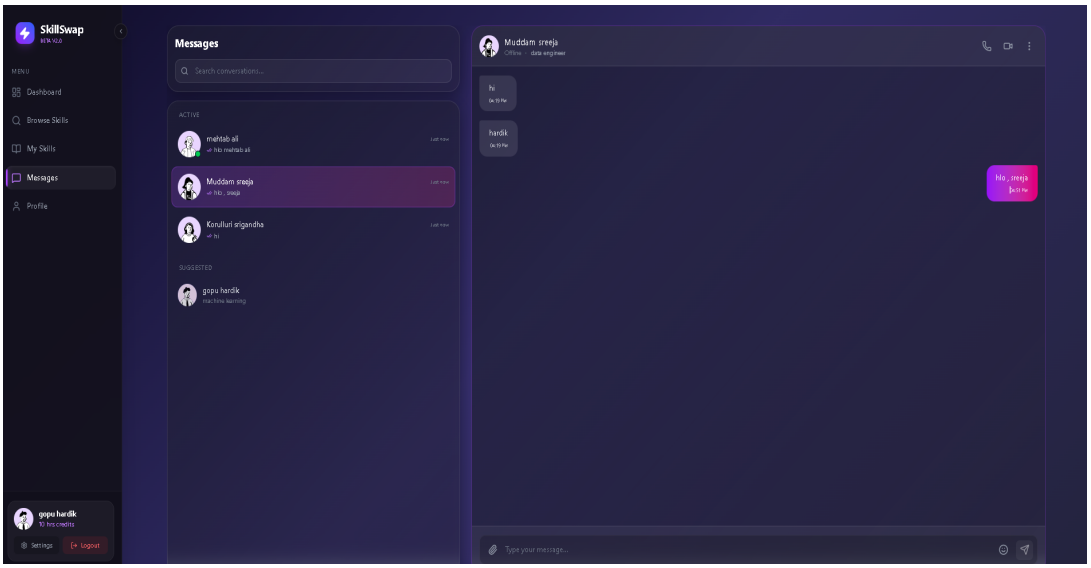

Figure 9: Messages – Real-time chat for skill exchange communication

9 Testing & Database Results

9.1 Testing Strategy

- **API Testing:** Tested individual endpoints using browser dev tools and Postman
- **Integration Testing:** Verified Frontend correctly displays data from Backend API
- **Cross-Browser Testing:** Checked UI consistency on Chrome, Edge, and Firefox

- **Database Testing:** Verified data persistence in MySQL Workbench

9.2 Test Results Summary

✓ Successfully implemented user authentication

✓ Skills stored and retrieved from MySQL database

✓ Chat messages persisted across sessions

✓ Real-time online/offline status tracking

✓ Responsive design on all screen sizes

9.3 Implementation Status

Table 8: Feature Implementation Status

Feature	**Status**
User Authentication	✓ Complete
Skill CRUD Operations	✓ Complete
Browse & Filter Skills	✓ Complete
Chat Messaging	✓ Complete
User Profiles	✓ Complete
Online Status Tracking	✓ Complete
Video Calls	△ UI Ready (WebRTC pending)
Rating System	○ Planned

9.4 Database Test Results

The following tables present actual data stored in the MySQL database, demonstrating successful implementation and data persistence.

9.4.1 Platform Statistics Overview

Table 9: Platform Statistics Summary (from rtrptotal.csv)

Metric	**Count**
Total Registered Users	4
Total Skills Listed	4
Total Messages Exchanged	7

9.4.2 Registered Users Data

Table 10: Users Table – Test Data from MySQL Database

Attribute	User 1	User 2	User 3	User 4
Username	gopu hardik	mehtab ali	Muddam sreeja	Korulluri srigandha
Email	gopuhardik @gmail.com	mehtabali @gmail.com	muddamsreeja18 @gmail.com	korullurisrigandha @gmail.com
Time Credits	10	10	10	10
Bio	I love to build and train LLMs	I love developing websites	I love exploring and analyzing data	I love programming
Role	Machine Learning	Web Development	Data Engineering	Python Programmer
Location	India	India	India	India
Is Online	Yes (1)	No (0)	No (0)	No (0)
Created At	2026-02-01 16:09	2026-02-01 16:12	2026-02-01 16:16	2026-02-01 16:20

9.4.3 Skills Marketplace Data

Table 11: Skills Table – Test Data from MySQL Database

ID	Title	Description	Category	Hours	User ID
1	Machine Learning	I can teach machine learning with real projects, including how to train and build models.	Programming	6	1
2	Web Development	I can teach web development, including frontend and backend technologies.	Programming	6	2
3	Data Engineering	I love to teach how to deal with data efficiently and at scale.	Programming	4	3
4	Python Developer	I can teach how to build applications and solve problems using the Python language.	Programming	3	4

9.4.4 Chat Messages Data

Table 12: Messages Table – Test Data from MySQL Database

ID	Sender	Receiver	Message	Timestamp
1	3	1	hi	2026-02-01 16:19:35
2	3	1	hardik	2026-02-01 16:19:39
3	3	2	hlo	2026-02-01 16:20:08
4	4	3	hi, sreeja	2026-02-01 16:21:11
5	1	2	hlo mehtab ali	2026-02-01 16:51:21
6	1	3	hlo, sreeja	2026-02-01 16:51:32
7	1	4	hi	2026-02-01 16:51:37

9.4.5 Data Verification Summary

The database test results confirm successful implementation:

- ✓ **User Registration:** 4 users registered with hashed passwords (bcrypt)
- ✓ **Profile Data:** All user profiles contain bio, role, and location
- ✓ **Skills Creation:** 4 skills with proper foreign key relationships
- ✓ **Messaging System:** 7 messages exchanged between users
- ✓ **Online Status:** is_online field correctly tracks user presence
- ✓ **Timestamps:** All records have accurate creation timestamps
- ✓ **Avatar Generation:** DiceBear API integration working

10 Challenges & Solutions

10.1 Personal Learnings

- Learned how to build a full-stack application with React and Node.js
- Understood JWT authentication flow and secure password handling
- Learned MySQL database design with foreign key relationships
- Implemented real-time features like online status tracking

10.2 Technical Challenges & Solutions

Table 13: Challenges and Solutions

Challenge	Solution
Database Migration	Migrated from SQLite to MySQL using mysql2 driver
State Management	Used React Context API for global state
API Authentication	Implemented JWT middleware for protected routes
Real-time Status	Added is_online column with login/logout tracking
Responsive Design	Used Tailwind CSS Grid and Flexbox

11 Future Scope

The following enhancements are planned for future development:

1. **Video Call Integration:** WebRTC-based in-app video conferencing for live skill sessions.
2. **Rating & Review System:** Allow learners to rate teachers and provide feedback.
3. **Skill Matching Algorithm:** AI-powered skill recommendations based on user preferences.
4. **Notification System:** Push notifications for messages and skill requests.
5. **Mobile Application:** React Native version for iOS and Android platforms.
6. **Payment Integration:** Optional paid premium skills for professional instructors.

12 Conclusion

SkillSwap serves as a fully functional proof-of-concept for a demonetized, community-driven education platform. The project successfully implements:

- ✓ Full-stack web development (React + Node.js + MySQL)
- ✓ Secure authentication with JWT
- ✓ Real-time features (chat, online status)
- ✓ Modern, responsive UI design
- ✓ Persistent data storage

The platform demonstrates the viability of peer-to-peer learning through time banking, providing a foundation for future expansion into a production-ready application.

References

1. React Documentation. https://react.dev
2. Node.js Official Documentation. https://nodejs.org

3. Express.js Guide. https://expressjs.com
4. MySQL Official Documentation. https://dev.mysql.com
5. JSON Web Tokens. https://jwt.io
6. Time Banking Concepts — Wikipedia and Research Papers
7. Tailwind CSS Documentation. https://tailwindcss.com
8. Vite Documentation. https://vitejs.dev

Acknowledgments

We would like to express our sincere gratitude to our mentor [Insert Mentor Name] for their guidance and support throughout this project. We also thank our institution for providing the resources and environment necessary for this development work.

Thank You!