Skill Swap

A skill exchange platform
(Project Report of DBAM)

Group Members:

ABDULLAH TAREEN – 210690
AYESHA SHAHZAD - 210680
AFSHEEN ABRAR – 210697

1. Executive Summary

Skill Swap is a full-stack web platform designed to enable users to exchange skills with each other in a structured and real-time manner. The system includes user registration, skill listing, skill swap requests, and real-time messaging once a swap is approved. An admin panel offers control over user, skill, and swap data. The system is built with modern technologies including React, Node.js, Socket.io, and MySQL.

2. Technical Architecture

2.1 Frontend (React.js + Vite)

- Routing: React Router
- Styling: Tailwind CSS
- State: Context API, React Query
- HTTP: Axios
- Real-time: Socket.io

2.2 Backend (Node.js + Express + MySQL)

- ORM: Sequelize
- Database: MySQL
- Authentication: JWT + Bcrypt
- Middleware: Auth handling, error catching
- Real-time: Socket.io

3. Database Design

3.1 Database Schema

Tables:

1. **Users Table:** Storing user's information like Name, email, password, role, id etc.
2. **Skills Table:** Consist of category of skill, expertise level and mapping with user's table via user id.
3. **Skill Swap Table:** Maintains records of swap requests send, what exchange is offering, approved or rejection status and mapping with user's table via user id.
4. **Chats Table:** Maintains chats records of users.

3.2 Database Relationships

- **One-to-Many:** Users to Skills
- **One-to-Many:** Users to Skill Swaps (as requester and requestee)
- **One-to-Many:** Skill Swaps to Messages
- **Many-to-Many:** Skills to Skills (through Skill Swaps)

3.3 Database Indexes

- Users: username, email, role
- Skills: userId, email, category
- Skill Swaps: requestedSkillId, offeredSkillId, requesterId, requesteeId, status
- Messages: skillSwapId, senderId, timestamp

4. Functional Modules / Features

4.1 User Authentication

- Registration form with fields:
  1. First Name
  2. Last Name
  3. Username (3-30 characters)
  4. Email (valid email format)
  5. Password (hashed with bcrypt)

- Login form with:
  1. Email/Username
  2. Password

- JWT-based authentication
- Protected routes for authenticated users

4.2 Skill Management

- Skill creation form with fields:
  1. Skill Name (1-100 characters)
  2. Description (optional)
  3. Category (Programming, Design, Language, Music, Other)
  4. Level (Beginner, Intermediate, Advanced, Expert)

- Skill listing with:
  1. Category-based grouping
  2. Visual indicators for skill levels
  3. Search and filter capabilities

- User can view all skills on the home page

4.3 Skill Swap System

- **Browsing Available Skills**
  - User can view all skills on the home page
  - Filter options available by: Category, Skill level

- **Initiating a Skill Swap**
  - User selects a skill they want to learn
  - Clicks "Request Swap" button
  - Selects one of their own skills to offer in exchange
  - Adds optional message explaining their interest
  - System notifies the skill owner

- **Swap Request Management**
  - Skill owner receives notification
  - Can view request details: Offered skill, Skill Level, Requester's profile

- **Message**
  - Options to: Accept the swap, Reject the swap

- **Post-Acceptance Process**
  - Upon acceptance:
    1. Chat feature is automatically enabled
    2. Both users receive confirmation
    3. Swap status updates to "Active"

- **Status management:**
  1. Pending
  2. Accepted
  3. Rejected
  4. Completed

4.4 Live Chat

- Real-time messaging using Socket.io
- Chat features:
  1. Message history
  2. Real-time message delivery
  3. Timestamp display

- Chat access restricted to accepted skill swaps only.

4.5 Admin Panel

- **User Management:**
  1. View all users
  2. Edit user details
  3. Monitor user activities
  4. Manage user permissions

- **Skills Management:**
  1. View all skills
  2. Edit skill details
  3. Remove inappropriate skills

- **Swap Management:**
  1. View all swaps
  2. Monitor swap status
  3. Resolve disputes

5. Security Considerations

- **Password hashing:** Bcrypt implementation for secure password storage and comparison.
- **JWT-based authentication:** Secure token-based authentication system with user roles and automatic refresh mechanism.
- **CORS protection:** Strict origin policies allowing only specific domains with controlled methods and headers.
- **Input validation:** Comprehensive validation at both frontend and backend for all user inputs.

- **SQL Injection Prevention:** Sequelize ORM implementation with parameterized queries and automatic sanitization.
- **Role-based Access:** Two-tier role system (user/admin) with middleware protection for routes.

6. Performance Considerations

- **Database Indexing:** Strategic indexes on frequently queried fields for optimized query performance.
- **React Query:** Efficient data fetching with automatic caching and background updates.
- **Socket.io:** Real-time communication implementation for instant messaging and notifications.
- **Query Optimization:** Efficient database queries with proper joins and selective field fetching.

7. Conclusion

Skill Swap successfully implements a comprehensive skill exchange platform with robust features for user interaction, skill management, and real-time communication. The application demonstrates strong database design principles and modern frontend development practices, making it a scalable and maintainable solution for skill exchange.

8. Instructions

To Run Frontend

```
cd frontend
npm install
npm run dev
```

To Run Backend

```
cd backend
npm install
node index.js
```

Env Setup Required

.env for backend must include:

- DB_USER
- DB_PASS
- DB_NAME
- JWT_SECRET
- SOCKET_PORT