**1. Base Task**

A skill-swap website where users can upload tutorials for others to view and learn. The website will support user registration, tutorial creation, rating, commenting, and messaging between users.

**2. Entities & Attributes**

1. **Users**
   * user\_id (int, primary key, auto-increment)
   * name (varchar(255))
   * email (varchar(255), unique)
   * password (varchar(255), hashed)
   * role (enum('teacher', 'student', 'admin'))
2. **Tutorials**
   * tutorial\_id (int, primary key, auto-increment)
   * user\_id (int, foreign key referencing users.user\_id)
   * title (varchar(255))
   * description (text)
   * created\_at (datetime, default current timestamp)
3. **Ratings**
   * rating\_id (int, primary key, auto-increment)
   * tutorial\_id (int, foreign key referencing tutorials.tutorial\_id)
   * user\_id (int, foreign key referencing users.user\_id)
   * rating (int, range 1-5)
4. **Comments**
   * comment\_id (int, primary key, auto-increment)
   * tutorial\_id (int, foreign key referencing tutorials.tutorial\_id)
   * user\_id (int, foreign key referencing users.user\_id)
   * comment (text)
   * created\_at (datetime, default current timestamp)
5. **Messages**
   * message\_id (int, primary key, auto-increment)
   * sender\_id (int, foreign key referencing users.user\_id)
   * receiver\_id (int, foreign key referencing users.user\_id)
   * message (text)
   * sent\_at (datetime, default current timestamp)

**3. Users Task/Access Definition**

1. **Users**
   * Can register, log in, and manage their profile.
   * Can create, edit, and delete their tutorials.
   * Can rate (1-5) and comment on other users' tutorials.
   * Can send messages to other users via a chat system.
   * Non-registered visitors can only view tutorials.
2. **Tutorials**
   * Created by users.
   * Display the creator's username, title, description, ratings, and comments.
3. **Messages/Chat**
   * Users can send and receive messages.
   * Messages are stored with sender and receiver IDs.
4. **Admin**
   * Can view all users and tutorials.
   * Can delete users or tutorials.
   * Deleting a user will also delete their tutorials, ratings, comments, and messages.

**4. Web Pages Structure**

1. **Home (index.php)**
   * Login and Register buttons.
   * List of all tutorials (no CRUD access for non-registered users).
2. **User Account (users.php)**
   * Display user profile (name, email).
   * Buttons to create tutorials and view tutorials (separate sections for self and others).
   * Self tutorials: Edit and delete options.
   * Others' tutorials: Rating and comment options.
3. **Create Tutorial (create\_tutorials.php)**
   * Form to input title and description.
4. **Edit Tutorial (edit\_tutorials.php)**
   * Similar to create\_tutorials.php, but pre-filled with existing tutorial data.
5. **Message (chat.php)**
   * Text area to write messages and a send button.
6. **Login (login.php)**
   * Form for email and password.
   * Error messages: "Incorrect Password!" or "User Doesn't Exist!".
7. **Register (register.php)**
   * Form for name, email, password, and role (teacher or student).
   * Error message: "User Already Exists!".
8. **Admin (admin.php)**
   * Display all users and tutorials.
   * Delete buttons for users and tutorials.

**Database Design**

Here’s the updated SQL schema:

sql

CREATE TABLE users (

user\_id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(255) NOT NULL,

email VARCHAR(255) UNIQUE NOT NULL,

password VARCHAR(255) NOT NULL,

role ENUM('teacher', 'student', 'admin') NOT NULL

);

CREATE TABLE tutorials (

tutorial\_id INT AUTO\_INCREMENT PRIMARY KEY,

user\_id INT,

title VARCHAR(255) NOT NULL,

description TEXT NOT NULL,

created\_at DATETIME DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (user\_id) REFERENCES users(user\_id) ON DELETE CASCADE

);

CREATE TABLE ratings (

rating\_id INT AUTO\_INCREMENT PRIMARY KEY,

tutorial\_id INT,

user\_id INT,

rating INT CHECK (rating BETWEEN 1 AND 5),

FOREIGN KEY (tutorial\_id) REFERENCES tutorials(tutorial\_id) ON DELETE CASCADE,

FOREIGN KEY (user\_id) REFERENCES users(user\_id) ON DELETE CASCADE

);

CREATE TABLE comments (

comment\_id INT AUTO\_INCREMENT PRIMARY KEY,

tutorial\_id INT,

user\_id INT,

comment TEXT NOT NULL,

created\_at DATETIME DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (tutorial\_id) REFERENCES tutorials(tutorial\_id) ON DELETE CASCADE,

FOREIGN KEY (user\_id) REFERENCES users(user\_id) ON DELETE CASCADE

);

CREATE TABLE messages (

message\_id INT AUTO\_INCREMENT PRIMARY KEY,

sender\_id INT,

receiver\_id INT,

message TEXT NOT NULL,

sent\_at DATETIME DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (sender\_id) REFERENCES users(user\_id) ON DELETE CASCADE,

FOREIGN KEY (receiver\_id) REFERENCES users(user\_id) ON DELETE CASCADE

);

**Frontend (HTML/CSS)**

1. **Home Page (index.php)**
   * Navigation bar with Login and Register buttons.
   * List of tutorials with title and description.
2. **User Account (users.php)**
   * Profile section (name, email).
   * Buttons for creating tutorials and viewing tutorials.
   * Separate sections for self and others' tutorials.
3. **Create/Edit Tutorial (create\_tutorials.php / edit\_tutorials.php)**
   * Form with fields for title and description.
4. **Message (chat.php)**
   * Text area and send button.
5. **Login (login.php)**
   * Form for email and password.
6. **Register (register.php)**
   * Form for name, email, password, and role.
7. **Admin (admin.php)**
   * Tables displaying users and tutorials.
   * Delete buttons for users and tutorials.

**Backend (PHP - Object-Oriented)**

1. **Database Connection (db.php)**
   * Create a class for database connection using PDO.
2. **User Class (User.php)**
   * Methods for registration, login, and profile management.
3. **Tutorial Class (Tutorial.php)**
   * Methods for creating, editing, deleting, and fetching tutorials.
4. **Rating/Comment Class (RatingComment.php)**
   * Methods for adding ratings and comments.
5. **Message Class (Message.php)**
   * Methods for sending and fetching messages.
6. **Admin Class (Admin.php)**
   * Methods for managing users and tutorials.