



User Management System with Express.js and MySQL

Objective

Students will build a simple web application using Node.js (Express) and MySQL that allows users to sign up, log in, and perform CRUD operations on a resource (like posts, tasks, or products).

Project Requirements

1. Backend: Node.js with Express.js
2. Database: MySQL
3. Functionality:
 - User signup (create account)
 - User login (authentication)
 - CRUD operations for a resource (Create, Read, Update, Delete)
4. Security: Store passwords securely using hashing (e.g., bcrypt)

Database Schema (Example)

Users Table:

| Column | Type | Constraints |
|------------|--------------|-----------------------------|
| id | INT | PRIMARY KEY, AUTO_INCREMENT |
| username | VARCHAR(50) | UNIQUE, NOT NULL |
| email | VARCHAR(100) | UNIQUE, NOT NULL |
| password | VARCHAR(255) | NOT NULL |
| created_at | TIMESTAMP | DEFAULT CURRENT_TIMESTAMP |

Posts Table (CRUD Resource Example):

| Column | Type | Constraints |
|--------|------|--------------|
| id | INT | PRIMARY KEY, |

| | | |
|------------|--------------|------------------------------|
| | | AUTO_INCREMENT |
| user_id | INT | FOREIGN KEY -> users(id) |
| title | VARCHAR(255) | NOT NULL |
| content | TEXT | |
| created_at | TIMESTAMP | DEFAULT CURRENT_TIMESTAMP |

Tasks for Students

1. Setup:

- Initialize Node.js project (npm init -y)
- Install dependencies: express, mysql2, dotenv, bcrypt, jsonwebtoken, body-parser

2. Database Connection:

- Connect to MySQL using mysql2

3. User Routes:

- POST /signup → create new user
- POST /login → authenticate user

4. CRUD Routes for Posts:

- POST /posts → create a new post
- GET /posts → get all posts
- GET /posts/:id → get a specific post
- PUT /posts/:id → update a post
- DELETE /posts/:id → delete a post

