

Project 1 — Users API Tutorial

Overview

- Stack: **PHP 8 + MySQL**
- Authentication: **JWT (Bearer)**
- CRUD: `/users`
- 10 REST APIs implemented

Endpoints (Public)

- `POST /auth/register`
- `POST /auth/login`
- `GET /users`
- `GET /users/{id}`
- `GET /users?q=...`

Endpoints (Bearer Protected)

- `POST /users` (*admin*)
- `PUT /users/{id}` (*self or admin*)
- `DELETE /users/{id}` (*admin*)
- `GET /users/me`
- `PUT /users/me`

Database Schema

- **Table:** `users`
- `id` BIGINT AUTO_INCREMENT PRIMARY KEY
- `name` VARCHAR(100)
- `email` VARCHAR(120) UNIQUE
- `password_hash` VARCHAR(255)
- `role` ENUM('student','instructor','admin')
- `created_at` TIMESTAMP

Registration Flow

Request:

```
POST /auth/register
{
  "name": "Ada",
  "email": "ada@example.com",
  "password": "P@ssw0rd"
}
```

Response:

```
201 Created
{
  "id": 1,
  "name": "Ada",
  "email": "ada@example.com",
  "role": "student"
}
```

Login Flow

Request:

```
POST /auth/login
{
  "email": "ada@example.com",
  "password": "P@ssw0rd"
}
```

Response:

```
200 OK
{
  "token": "<JWT>",
  "user": {...}
}
```

Bearer Token

- Authorization: Bearer <JWT>
- Expiration: 4 hours
- Roles: student, instructor, admin
- Required for secure routes

GET /users

- Public endpoint
- Returns all users (no passwords)
- Supports search `?q=`

```
curl http://127.0.0.1:8080/users
```


GET /users/{id}

- Public endpoint
- Returns single user

```
curl http://127.0.0.1:8080/users/1
```

POST /users (*admin*)

- Create new user (Bearer required)

```
curl -X POST /users \  
  -H "Authorization: Bearer $TOKEN" \  
  -d '{"name":"Bob","email":"bob@example.com","password":"pw","role":"student"}'
```

PUT /users/{id}

- Update user (self or admin)

```
curl -X PUT /users/1 \  
  -H "Authorization: Bearer $TOKEN" \  
  -d '{"name":"Ada Updated"}'
```

DELETE /users/{id}

- Delete user (*admin only*)

```
curl -X DELETE /users/1 -H "Authorization: Bearer $TOKEN"
```

GET /users/me

- Returns info of logged-in user

```
curl /users/me -H "Authorization: Bearer $TOKEN"
```

PUT /users/me

- Update logged-in user's info

```
curl -X PUT /users/me \  
  -H "Authorization: Bearer $TOKEN" \  
  -d '{"name":"New Name"}'
```

Security Features

- Bcrypt password hashing
- JWT (HS256)
- Role-based access
- 401/403 handling

Testing Tools

- `code/tests/curl_all.sh`
- `code/public/tester.html`
- Manual curl and browser testing

Error Handling

- 400: Validation error
- 401: Missing or invalid token
- 403: Forbidden
- 404: Not found
- 409: Email exists
- 500: Database error

Rubric Coverage

- 10+ endpoints
- 2+ Bearer routes
- MySQL CRUD
- Testers
- Slides

Project Completed

GitHub: github.com/Abdullahi5670/project-1