

Next.js 14 Authentication Dashboard with MySQL Backend

1. Overview

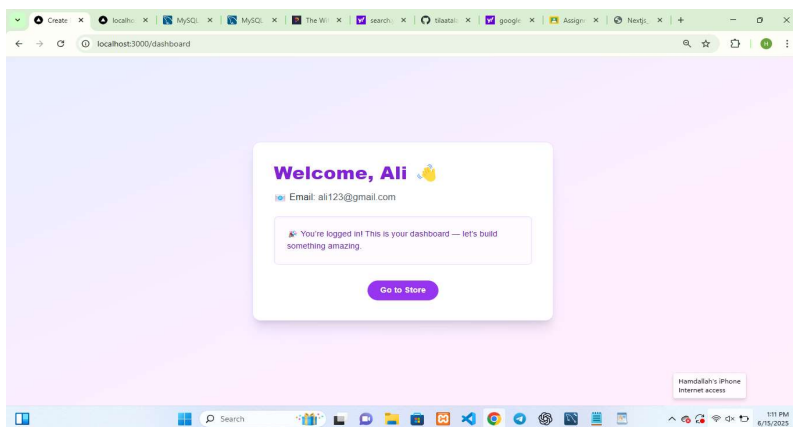
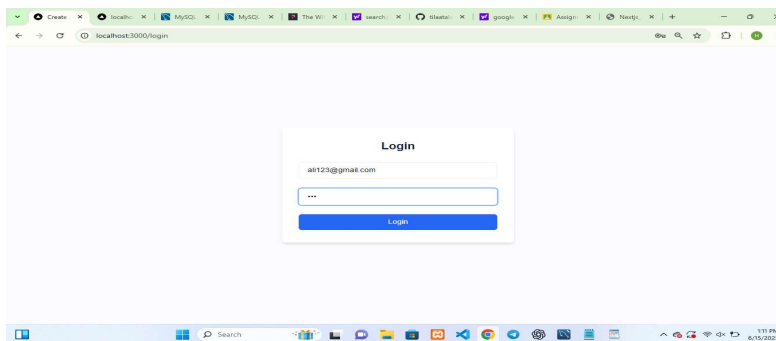
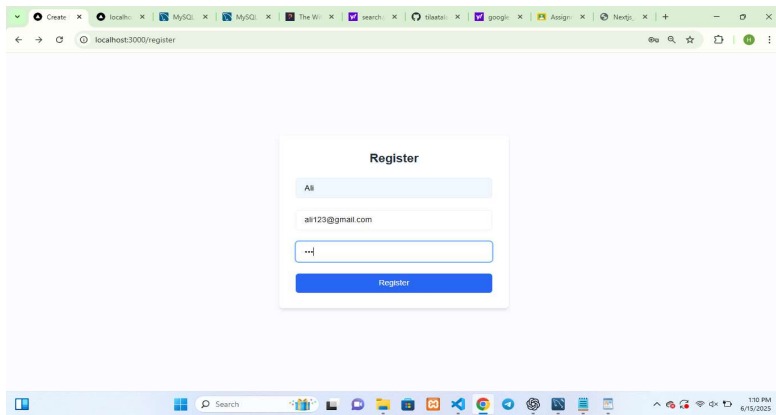
This project is a simple authentication system built using Next.js 14 and a MySQL database. It includes user registration, login, and a protected dashboard that displays user information. The purpose is to demonstrate full-stack development skills with secure authentication and proper database integration.

2. Tools & Technologies

- Frontend: Next.js 14 (App Router)
- Backend: Node.js (API Routes in Next.js)
- Database: MySQL
- Authentication: bcrypt, JWT
- UI: Tailwind CSS

3. Features Implemented

- Register Page: New users can sign up.
- Login Page: Authenticates existing users.
- Password Security: Passwords are hashed using bcrypt.
- JWT Cookies: Token stored after login.
- Protected Dashboard: Displays user info only if logged in.
- API Routes: Built-in API logic for login, registration, and user data.



4. Database Schema

CREATE TABLE users (

id INT NOT NULL AUTO_INCREMENT,

```
username VARCHAR(256) NOT NULL,  
  
email VARCHAR(256) NOT NULL,  
  
pass_word VARCHAR(255),  
  
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
  
PRIMARY KEY (id),  
  
UNIQUE (email)  
  
);
```

The users table stores user information, including a hashed password and account creation timestamp.

5. Scripts Used

- db/schema.sql: Contains SQL to create the users table.
- db/backup.sql: Full MySQL dump of the database and existing user data.

6. How to Run the Project

- Clone the GitHub repo.
- Run npm install.
- Create a .env.local file with:

DB_HOST=localhost

DB_USER=root

DB_PASSWORD=yourpassword

DB_NAME=nextjs_db

JWT_SECRET=your_jwt_secret
- Start the server: npm run dev
- Visit: <http://localhost:3000>

7. Notes

- Uses a local MySQL database.
- Passwords in the backup are hashed.
- JWT cookies are HttpOnly.
- Dashboard is a protected route.