

User Authentication System Documentation

Table of Contents

- [Overview](#)
- [Tech Stack](#)
- [Features](#)
- [Setup Instructions](#)
- [Usage Guide](#)
- [API Endpoints](#)
- [Security Implementation](#)
- [Troubleshooting](#)
- [Project Structure](#)
- [Future Enhancements](#)
- [License](#)

Overview

This system provides a comprehensive user authentication solution with registration, login, and profile display capabilities. User passwords are securely hashed with bcrypt before storage in a PostgreSQL database. The frontend features responsive design with modern animations.

Tech Stack

- **Frontend:** HTML, CSS, JavaScript
- **Backend:** Node.js with Express
- **Database:** PostgreSQL
- **Security:** bcrypt for password hashing

Features

- User registration with validation
- Secure login with password hashing
- User dashboard with profile info
- Form animations and responsive design
- PostgreSQL integration for data persistence
- Password visibility toggle for better UX

Setup Instructions

Prerequisites

- Node.js (v16+ recommended)
- npm (comes with Node.js)
- PostgreSQL (running locally)
- Git (for cloning the repo)

Installation Steps

1. Clone the Repository

```
git clone https://github.com/your-username/your-repo.git
cd your-repo/user-auth
```

2. Install Backend Dependencies

```
cd backend
npm install express cors pg bcrypt
```

3. Database Setup

- Start your PostgreSQL server
- Create a database named `auth_db`:

```
CREATE DATABASE auth_db;
```

- Create the users table:

```
CREATE TABLE users (  
  id SERIAL PRIMARY KEY,  
  first_name VARCHAR(100) NOT NULL,  
  last_name VARCHAR(100) NOT NULL,  
  email VARCHAR(100) UNIQUE NOT NULL,  
  date_of_birth DATE NOT NULL,  
  password VARCHAR(255) NOT NULL  
);
```

4. Configure Backend

- Update database credentials in `backend/server.js` if needed
- Default configuration:

```
const pool = new Pool({  
  user: "postgres",  
  host: "localhost",  
  database: "auth_db",  
  password: "your_postgres_password",  
  port: 5432,  
});
```

5. Start the Backend Server

```
node server.js

- Server runs on http://localhost:8080
```

6. Launch the Frontend

- Open `login.html` in a browser
- Or use VS Code's Live Server extension

Usage Guide

Registration

1. Navigate to registration page from login screen
2. Fill out all required fields (first name, last name, email, date of birth, password)
3. Passwords must match and meet minimum requirements
4. Submit the form
5. On successful registration, you'll be redirected to login page

Login

1. Enter your registered email address
2. Enter your password (toggle visibility if needed)
3. Click "Sign In"
4. On successful login, you'll be redirected to dashboard

Dashboard

1. View your profile information (name, email, date of birth)
2. Use the logout button to end your session

API Endpoints

User Registration

- **URL:** `/api/users/register`
- **Method:** POST
- **Request Body:**

```
{
  "firstName": "BARATH",
  "lastName": "G",
```

```
"email": "barath@example.com",  
"dateOfBirth": "2007-01-17",  
"password": "securepassword"  
}
```

-Response: **201** Created or appropriate error

User Login

- **URL:** `/api/users/login`
- **Method:** POST
- **Request Body:**

```
{  
  
  "email": "barath@example.com",  
  "password": "securepassword"  
}
```

-Response: **200** OK or appropriate error

User Profile

- **URL:** `/api/users/profile?email=barath@example.com`
- **Method:** GET
- **Response :** User information as JSON

**

Note the important correction: the "**Response**" line should be a bullet point with formatting like the other lines, not an indented line with a single hyphen .**

Security Implementation

- **Password Hashing:** All passwords are hashed using bcrypt with 10 salt rounds
- **Input Validation:** Client and server-side validation prevents malicious inputs
- **CORS Protection:** API configured with CORS to control domain access

Troubleshooting

- **Database Connection Errors:** Verify PostgreSQL is running and credentials are correct
- **Registration Failures:** Check if email is already registered
- **Login Issues:** Ensure email and password match a registered account
- **Backend Not Starting:** Check for port conflicts on 8080

Project Structure

```
user-auth/  
├── auth.js           # Frontend authentication logic  
├── dashboard.html    # User dashboard page  
├── index.js          # Main JavaScript file  
├── login.html        # Login page  
├── package.json      # Frontend dependencies  
├── register.html     # Registration page  
├── styles.css        # CSS styling  
└── backend/  
    ├── package.json  # Backend dependencies  
    ├── README.md     # Project documentation  
    └── server.js     # Express server with API endpoints
```

This structure separates frontend and backend components, with the backend folder containing server code and database interactions, while the root directory contains all frontend files.

Future Enhancements

Short-term Improvements

- **Email Verification:** Add email verification step during registration
- **Password Reset:** Implement "Forgot Password" functionality
- **Remember Me:** Add persistent login with secure cookies

Mid-term Goals

- **OAuth Integration:** Allow login with Google, Facebook, or GitHub
- **User Roles:** Implement role-based access control (admin, user)
- **Account Management:** Allow users to update profile information

Long-term Vision

- **Two-Factor Authentication:** Add 2FA for enhanced security
- **API Rate Limiting:** Prevent brute force attacks
- **Activity Logging:** Track login attempts and user actions

License

This project is released into the public domain and is provided "as is" without warranty or copyright claims:

This is free and unencumbered software released into the public domain.

Anyone is free to copy, modify, publish, use, compile, sell, or distribute this software, either in source code form or as a compiled binary, for any purpose, commercial or non-commercial, and by any means.

In jurisdictions that recognize copyright laws, the author or authors of this software dedicate any and all copyright interest in the software to the public domain. We make this dedication for the benefit of the public at large and to the detriment of our heirs and successors. We intend this dedication to be an overt act of relinquishment in perpetuity of all present and future rights to this software under copyright law.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For more information, please refer to <<https://unlicense.org>>