



# HEALTH PROGRAM API TEST

Software Engineering Intern  
Application at CEMA

Presented By **Dennis Wambua**

 **URL-API**

# PROJECT OVERVIEW

## OVERVIEW

### What is it?

A backend system built with Node.js, Express, and MongoDB.

### Purpose

Manage client registrations and health programs.

Provide secure authentication using JWT.

Allow admins to manage programs and clients to enroll in them.

# Features

## Data Protection with Password Hashing

- Secure Password Storage:
  - Passwords are hashed using bcrypt before being stored in the database.
  - This ensures that even if the database is compromised, the actual passwords remain protected.
- Password Verification:
  - During login, the provided password is compared with the hashed password using bcrypt's compare method.
  - This ensures that only valid credentials can access the system.
- Why Bcrypt?:
  - Bcrypt adds a computational cost to hashing, making brute-force attacks significantly harder.
  - It automatically handles salting, adding an extra layer of security to the hashing process.

## Client Management:

- Register new clients: Clients can register by providing their details, including a password.
  - Login and receive a JWT token: Clients can log in using their credentials to receive a secure token for authentication.
  - Search for clients by name or national ID: Admins can search for clients using flexible filters.
  - View client profiles and enrolled programs: Retrieve detailed client information, including their enrolled programs.
- 

## Program Management:

- Admins can create health programs: Admin users can create and manage health programs.
  - Clients can enroll in one or more programs: Clients can enroll in programs to track their participation
- 

## Authentication & Authorization:

- JWT-based authentication: Securely authenticate users with JSON Web Tokens (JWT).
  - Admin-only access for program creation: Restrict program creation to admin users using middleware.
-

# Technologies Used

## Backend

Node.js, Express.js

---

## Database

MongoDB + Mongoose  
ODM

---

## Authentication

JSON Web Tokens (JWT)

---

## Environment Configuration

dotenv

---

## Dev Tools & Testing

ESLint, Nodemon, Babel,  
Jest + Supertest

# API ENDPOINTS

Method	Endpoint	Description	Auth Required
POST	/api/clients	Register a new client	✗ No
POST	/api/clients/login	Login and receive JWT token	✗ No
GET	/api/clients	Search clients by name/national ID	✓ Yes
GET	/api/clients/:id	Get a specific client's profile	✓ Yes
POST	/api/clients/:id/enrollments	Enroll a client in health programs	✓ Yes

Method	Endpoint	Description	Auth Required	Admin Only
GET	/api/programs	Create a new program	✓ Yes	✓ Yes

Back to Agenda

# Middleware & Testing

## AUTHENTICATION MIDDLEWARE

- Validates JWT token and attaches user info to req.user.
- Protects private endpoints.

## ADMIN MIDDLEWARE

- Restricts access to admin-only features like program creation.

## TESTING

- Ensure API endpoints work as expected.
- Validate authentication and authorization flows.



# Thank You

EMAIL

denny0912dw@gmail.com

