Enhanced Security:

Added Helmet for security headers

Implemented rate limiting

Stronger password validation

Better JWT implementation

Improved error handling

Better Validation:

Comprehensive Joi validation with custom error messages

MongoDB schema-level validation

Input sanitization

Proper error handling for all cases

Improved Error Handling:

Centralized error handling middleware

Detailed error messages

Proper HTTP status codes

Development/production error responses

API Endpoints:

Register (POST /api/users/register):

{

"username": "user",

"password": "Password123!",

"email": "user@example.com",

"fullName": "user Clint",

"gender": "male",

"dateOfBirth": "2005-01-01",

"country": "INDIA"

}

**Login** (POST /api/auth/login):

{

"username": "user",

"password": "Password123!"

}

**Search** (GET /api/users/search?query=john):

Requires Bearer token in Authorization header

Returns matching users (excludes passwords)

To test in Postman:

Create a new collection

Add the three endpoints

For the search endpoint, add an Authorization header with the token from login

Test the flow: register → login → search

The system now includes:

Strong input validation

Proper error handling

Security best practices

Rate limiting

Logging

Better MongoDB integration

Improved code organization