API Documentation – User Authentication & Job Search Backend

1. POST /api/signup

Creates a new user account with profile information, hashed password, and skill preferences.

Request Body (JSON)

- "username": "jsmith"
- "password": "Password1"
- "firstName": "John"
- "lastName": "Smith"
- "middleInitial": "A"
- "email": "john@example.com"
- "phone": "555-555-5555"
- "city": "Portland"
- "county": "Cumberland"
- "zip": "04101"
- "maxCommute": 30
- "remotePref": "Hybrid"
- "salaryMin": 50000
- "salaryMax": 90000
- "skills": "python, sql, communication"

Success Response

{ "success": true, "message": "Signup successful!", "user_id": 27 }

Error Codes

- 400 – Username or email already exists
- 500 – Server/database error

2. POST /api/auth/login

Authenticates the user using bcrypt password comparison.

Request Body

{ "username": "jsmith", "password": "Password1" }

Success Response

{ "success": true, "user": { ... }}

Error Codes

- 400 – Missing username or password
- 401 – Invalid username or password

- 500 – Server error

3. GET /api/jobs/search

Performs a keyword search across job titles, descriptions, and employer names.

Query Parameter

keyword

Example Request

GET /api/jobs/search?keyword=developer

Success Response

{ "success": true, "jobs": [...] }

4. GET /api/jobs/filter

Returns jobs ranked using stored procedure jobMatchScoreFilterSearch.

Query Parameters

- city
- remote
- min_salary
- max_salary
- skills (optional)

Example Request

GET /api/jobs/filter?city=Portland&remote=Hybrid&min_salary=50000&skills=python,sql

Success Response

{ "success": true, "jobs": [...] }

5. GET /api/jobs/recommend/

Returns recommended jobs for a user based on stored procedure jobMatchScore.

Example Request

GET /api/jobs/recommend/27

Success Response

{ "success": true, "jobs": [...] }