



SkillMatch API Documentation

This documentation details the available API endpoints for the SkillMatch backend application.

Base API: <https://skillmatch-ej4r.onrender.com>

Swagger Doc: [Swagger UI](#)

🔑 Authentication Routes

Base Path: /api/auth

Method	Endpoint	Description	Access
POST	/signup	Registers a new user account.	Public
POST	/verify	Activates a new account using a verification code.	Public
POST	/resendCode	Requests a new account verification code.	Public
POST	/login	Logs in a user and returns a JWT token.	Public
POST	/forgotPassword	Initiates the password reset process by sending a code to the user's email.	Public
POST	/verifyResetCode	Verifies the password reset code.	Public
PATCH	/resetPassword	Sets a new password using a valid reset code.	Public

POST /signup

Component	Details
Request Body	username, email, password, passwordConfirm, skills (optional)
Success (201)	status: "success", message: "Registration successful! The verification code is being sent to your email."
Fail (400)	Email Taken: status: "fail", message: "This email is already taken"
Fail (400)	Password Mismatch: status: "fail", message: "Passwords do not match."
Error (400/500)	status: "fail"/"error", message: err.message

POST /verify

Component	Details
Request Body	email, code (The 6-digit verification code)
Success (200)	status: "success", token, data: { user: {...} } (Returns a login token and user object)
Fail (400)	status: "fail", message: "The code is invalid or has expired."
Error (500)	status: "error", message: err.message

POST /login

Component	Details
Request Body	email, password
Success (200)	status: "success", token, data: { user: {...} }
Fail (400)	Missing Fields: status: "fail", message: "Please provide email and password"
Fail (401)	Invalid Credentials: status: "fail", message: "Invalid email or password"

Component	Details
Fail (401)	Not Verified: status: "fail", message: "Account is not verified. Please verify your account first."
Error (500)	status: "error", message: err.message

POST /forgotPassword

Component	Details
Request Body	email
Success (200)	status: "success", message: "If the user exists, a password recovery code is being sent." (Returns 200 even if the user doesn't exist for security reasons)
Error (500)	status: "error", message: err.message

PATCH /resetPassword

Component	Details
Request Body	email, code, newPassword, passwordConfirm
Success (200)	status: "success", token, message: "Password reset successfully. You are now logged in." (Logs the user in)
Fail (400)	Password Mismatch: status: "fail", message: "New passwords do not match."
Fail (400)	Invalid/Expired Code: status: "fail", message: "The code is invalid or has expired."
Error (500)	status: "error", message: err.message

User Profile Routes

Base Path: /api/auth (All protected by authMiddleware.protect)

Method	Endpoint	Description	Access
GET	/me	Get the profile data of the currently logged-in user.	Protected
PATCH	/updateMe	Update non-password user information (username, email, bio, location, skills, interests).	Protected
PATCH	/updateMyPassword	Change the user's password.	Protected
PATCH	/updateMyPhoto	Upload/update the user's profile photo.	Protected
GET	/me/saved	Get the list of opportunities the user has saved.	Protected
PATCH	/updateSkillsAndInterests	Update user's skills, interests, and/or bio.	Protected
DELETE	/deleteMe	Deactivate the currently logged-in user's account (active: false).	Protected

PATCH /updateMe

Component	Details
Request Body	Any of: username, email, bio, location, skills, interests.
Success (200)	status: "success", data: { user: {...} }
Fail (400)	Password Attempt: status: "fail", message: "This link is not for changing your password. Use /updateMyPassword"

Component	Details
Error (400)	status: "fail", message: err.message (e.g., validation error)

PATCH /updateMyPassword

Component	Details
Request Body	passwordCurrent, password, passwordConfirm
Success (200)	status: "success", token, message: "Password updated successfully. You are now logged in with the new token."
Fail (401)	Incorrect Current Password: status: "fail", message: "The current password is incorrect."
Fail (400)	New Password Mismatch: status: "fail", message: "New passwords do not match."
Error (500)	status: "error", message: err.message

GET /me/saved

Component	Details
Success (200)	status: "success", results, data: { savedOpportunities: [...] } (Opportunities are populated with fields: title, company, requiredSkills, tags, salary, level, type)
Error (400)	status: "fail", message: err.message

💼 Opportunity Routes (Public/User)

Base Path: /api/opportunities

Method	Endpoint	Description	Access
GET	/	Get a list of all opportunities (supports filtering, sorting, pagination, and searching).	Public
GET	/:id	Get details for a specific opportunity.	Public
GET	/recommended	Get a list of opportunities recommended based on the user's skills and interests.	Protected
POST	/save/:id	Saves an opportunity to the user's profile.	Protected
DELETE	/unsave/:id	Removes an opportunity from the user's saved list.	Protected

GET / (Opportunities List)

Component	Details
Query Parameters	page, limit, sort, fields, search (Search by opportunity title) and other general query filters.
Success (200)	status: "success", results, total, page, data: { opportunities: [...] }
Error (400)	status: "fail", message: err.message

GET /:id (Single Opportunity)

Component	Details
Success (200)	status: "success", data: { opportunity: {...} }
Fail (404)	status: "fail", message: "No opportunity found with that ID"
Error (404)	status: "fail", message: err.message

GET /recommended

Component	Details
Success (200)	status: "success", results, data: { recommendedOpportunities: [...] } (Opportunities are scored and sorted by relevance)
Success (200)	No Keywords: status: "success", message: "No keywords found in your profile. Please update your skills and interests for personalized recommendations.", data: { recommendedOpportunities: [] }
Error (500)	status: "error", message: err.message

POST /save/:id

Component	Details
Path Parameter	id (Opportunity ID)
Success (200)	status: "success", message: "Opportunity saved successfully.", data: { savedCount: <number> }
Error (400)	status: "fail", message: err.message

👑 Admin Routes

Base Path: /api/admin

All routes under /api/v1/admin are **Protected** and require the user's role to be "**admin**" (authMiddleware.protect, then authMiddleware.authorize("admin")).

Admin Dashboard and Users

Method	Endpoint	Description
GET	/dashboard	Get summary data for the admin dashboard (total active users, total opportunities).
GET	/users	Get a list of all users.
PATCH	/users/:id	Update user data by ID (can update name, email, role, active, isVerified).
DELETE	/users/:id	Delete a user by ID.

PATCH /users/:id (Update User)

Component	Details
Request Body	Any of: name, email, role, active (boolean), isVerified (boolean). Cannot change password.
Success (200)	status: "success", data: { user: {...} }
Fail (400)	Password Attempt: status: "fail", message: "Admins cannot change user passwords using this route."
Fail (404)	status: "fail", message: "No user found with that ID."
Error (400)	status: "fail", message: err.message

Admin Opportunity Management

Method	Endpoint	Description
GET	/opportunities	Get a list of all opportunities (same as public route).
POST	/opportunities	Create a new opportunity.
GET	/opportunities/:id	Get a specific opportunity by ID (same as public route).
PATCH	/opportunities/:id	Update an existing opportunity by ID.

Method	Endpoint	Description
DELETE	/opportunities/:id	Delete an opportunity by ID.

POST /opportunities (Create Opportunity)

Component	Details
Request Body	Includes opportunity details (e.g., title, description, requiredSkills, tags, etc.). createdBy is automatically set to the admin's ID.
Success (201)	status: "success", data: { opportunity: {...} }
Error (400)	status: "fail", message: err.message (e.g., validation error)

PATCH /opportunities/:id (Update Opportunity)

Component	Details
Request Body	Any opportunity fields to be updated.
Success (200)	status: "success", data: { opportunity: {...} }
Fail (404)	status: "fail", message: "No opportunity found with that ID"
Error (400)	status: "fail", message: err.message

DELETE /opportunities/:id (Delete Opportunity)

Component	Details
Success (204)	status: "success", data: null (No content returned)
Fail (404)	status: "fail", message: "No opportunity found with that ID"
Error (500)	status: "error", message: err.message