# API DOCUMENTATION - ALIGN EYE

## **1. Send OTP**

### **Endpoint**

POST /api/v1/auth/sendotp

### **Description**

Sends a 6-digit numeric OTP to the given email address. This API is typically used before user registration to verify the email.

**Request**

#### **Headers**

* Content-Type: application/json

#### **Body**

json

CopyEdit

{

"email": "....@gmail.com"

}

**Responses**

#### **Success (200 OK)**

{

"success": true,

"message": "OTP Sent Successfully",

"otp": "407976"

}

#### **User Already Registered (401 Unauthorized)**

{

"success": false,

"message": "User is Already Registered"

}

#### **Server Error (500 Internal Server Error)**

{

"success": false,

"error": "Error message details"

}

**Authentication Required**

No

**Notes**

* OTP is a 6-digit numeric code.
* OTPs are stored in a MongoDB collection for verification.
* Duplicates are avoided by checking against existing OTPs in the database

## **2. Signup**

### **Endpoint**

POST /api/v1/auth/signup

### **Description**

Registers a new user after verifying their OTP. It ensures all required fields are provided, checks for existing users, validates the OTP, hashes the password, and creates a user and profile.

**Request**

#### **Body**

{

"firstName": "Jashan",

"lastName": "Singh",

"email": "jaskochar2003@gmail.com",

"password": "123456",

"confirmPassword": "123456",

"accountType": "Customer",

"contactNumber": 1233456789,

"otp": "943229"

}

**Responses**

#### **Success (200 OK)**

json

CopyEdit

{

"success": true,

"user": {

"firstName": "Jashan",

"lastName": "Singh",

"email": "jaskochar2003@gmail.com",

"password": "$2b$10$...",

"accountType": "Customer",

"active": true,

"approved": true,

"additionalDetails": "685696dda316c33c4c848c04",

"image": "https://api.dicebear.com/5.x/initials/svg?seed=Jashan Singh",

"provider": "local",

"googleId": null,

"\_id": "685696dda316c33c4c848c06",

"createdAt": "2025-06-21T11:26:21.913Z",

"updatedAt": "2025-06-21T11:26:21.913Z",

"\_\_v": 0

},

"message": "User registered successfully"

}

**Missing Fields (403 Forbidden)**

{

"success": false,

"message": "All Fields are required"

}

#### **Password Mismatch (400 Bad Request)**

{

"success": false,

"message": "Password and Confirm Password do not match. Please try again."

}

#### **User Already Exists (400 Bad Request)**

{

"success": false,

"message": "User already exists. Please sign in to continue."

}

#### **Invalid OTP (400 Bad Request)**

{

"success": false,

"message": "The OTP is not valid"

}

#### **Server Error (500 Internal Server Error)**

{

"success": false,

"message": "User cannot be registered. Please try again."

}

**Notes**

* OTP must be generated beforehand using the /sendotp endpoint.
* accountType should be either "Customer" or "Employee" or “Admin”.

## **3. Login**

### **Endpoint**

POST /api/v1/auth/login

### **Description**

Authenticates a user using email and password. On successful login, returns a JWT token, sets it in an HTTP-only cookie, and returns user details.

### **Request**

#### **Headers**

* Content-Type: application/json

#### **Body**

json

CopyEdit

{

"email": "jaskochar2003@gmail.com",

"password": "123456"

}

### **Responses**

### **Success (200 OK)**

{

"success": true,

"token": "<JWT\_TOKEN>",

"user": {

"\_id": "685696dda316c33c4c848c06",

"firstName": "Jashan",

"lastName": "Singh",

"email": "jaskochar2003@gmail.com",

"accountType": "Customer",

"active": true,

"approved": true,

"additionalDetails": {

"\_id": "685696dda316c33c4c848c04",

"gender": null,

"dateOfBirth": null,

"about": null,

"contactNumber": 1233456789,

"\_\_v": 0

},

"image": "https://api.dicebear.com/5.x/initials/svg?seed=Jashan Singh",

"provider": "local",

"googleId": null,

"createdAt": "2025-06-21T11:26:21.913Z",

"updatedAt": "2025-06-21T11:31:06.940Z",

"\_\_v": 0,

"token": "<JWT\_TOKEN>"

},

"message": "User Login Success"

}

**Missing Fields (400 Bad Request)**

{

"success": false,

"message": "Please Fill up All the Required Fields"

}

#### **User Not Found (401 Unauthorized)**

{

"success": false,

"message": "User is not Registered with Us Please SignUp to Continue"

}

#### **Incorrect Password (401 Unauthorized)**

{

"success": false,

"message": "Password is incorrect"

}

#### **Server Error (500 Internal Server Error)**

{

"success": false,

"message": "Login Failure Please Try Again"

}

**Authentication Required**

No

**Notes**

* On successful login:  
  + A **JWT token** is generated and stored both in the response and as an **HTTP-only cookie** valid for 3 days.
  + The user’s password is excluded from the response.
  + The user's additionalDetails (profile) is populated.
* Token contains: email, id, and accountType.

## **4. Update Profile**

### **Endpoint**

PUT /api/v1/profile/updateProfile

### **Description**

Updates a user's basic information (first name, last name) and profile details (DOB, about, gender, contact number). Requires authentication.

**Request**

#### **Headers**

* Content-Type: application/json
* Authorization: Bearer <JWT\_TOKEN>

#### **Body**

{

"firstName": "Jashan",

"lastName": "Singh",

"dateOfBirth": "2-9-2003",

"about": "My self Jashanpreet Singh , i am a student of Panjab University currently pursuing my degree in Computer Science",

"contactNumber": 1234567890,

"gender": "Male"

}

**Responses**

### **Success (200 OK)**

{

"success": true,

"message": "Profile updated successfully",

"updatedUserDetails": {

"\_id": "685696dda316c33c4c848c06",

"firstName": "Jashan",

"lastName": "Singh",

"email": "jaskochar2003@gmail.com",

"password": "$2b$10$...",

"accountType": "Customer",

"active": true,

"approved": true,

"additionalDetails": {

"\_id": "685696dda316c33c4c848c04",

"gender": "Male",

"dateOfBirth": "2-9-2003",

"about": "My self Jashanpreet Singh , i am a student of Panjab University currently pursuing my degree in Computer Science",

"contactNumber": 1234567890,

"\_\_v": 0

},

"image": "https://api.dicebear.com/5.x/initials/svg?seed=Jashan Singh",

"provider": "local",

"googleId": null,

"createdAt": "2025-06-21T11:26:21.913Z",

"updatedAt": "2025-06-21T11:36:01.411Z",

"\_\_v": 0,

"token": "<JWT\_TOKEN>"

}

}

**Server Error (500 Internal Server Error)**

{

"success": false,

"error": "Error message details"

}

**Authentication Required**

✅ Yes (JWT in Authorization header)

**Notes**

* The endpoint requires a valid JWT token to access the authenticated user's ID (req.user.id).
* Only fields sent in the request are updated. Others remain unchanged.
* Returns the full updated user object including populated profile.

**5. Create Exercise Category**

### **Endpoint**

POST /api/v1/exercise/createExerciseCategory

### **Description**

Creates a new exercise category by providing a name and description. Useful for organizing exercises into different categories (e.g., Mind, Body, Strength).

**Request**

#### **Body**

{

"name": "Mind",

"description": "Exercise which helps in releasing pressure from mind"

}

**Responses**

#### **Success (200 OK)**

{

"success": true,

"categorysDetails": {

"name": "Mind",

"description": "Exercise which helps in releasing pressure from mind",

"courses": [],

"\_id": "68569c38eed8a344c6a0ccb7",

"\_\_v": 0

},

"message": "Category Created Successfully"

}

#### **Missing Fields (400 Bad Request)**

{

"success": false,

"message": "All fields are required"

}

#### **Server Error (500 Internal Server Error)**

{

"success": false,

"message": "Error message details"

}

**Authorization Required**

Yes, user should be Admin

**Notes**

* courses is initialized as an empty array; presumably, exercises or modules can be added later under this category.
* Field validation ensures both name and description are mandatory.

**6. Show All Exercise Categories**

### **Endpoint**

GET /api/v1/exercise/showAllCategories

### **Description**

Fetches and returns all available exercise categories from the database.

**Request**

#### **Headers**

* No special headers required

#### **Body**

* **Not required** for this GET request

**Responses**

#### **Success (200 OK)**

{

"success": true,

"data": [

{

"\_id": "68569c38eed8a344c6a0ccb7",

"name": "Mind",

"description": "Exercise which helps in releasing pressure from mind",

"courses": [],

"\_\_v": 0

}

]

}

#### **Server Error (500 Internal Server Error)**

{

"success": false,

"message": "Error message details"

}

**Authentication Required**

No

**Notes**

* The response returns an array of all exercise categories in the system.
* Each category includes its ID, name, description, and associated courses (initially an empty array).

**7. Create Blogs Category**

### **Endpoint**

POST /api/v1/blogs/createCategory

### **Description**

Creates a new blog category by specifying a name and description. Used to group blogs into meaningful sections (e.g., Eye, Nutrition, Mental Health).

**Request**

#### **Headers**

* Content-Type: application/json

#### **Body**

{

"name": "Eye",

"description": "Blogs related to eyes"

}

**Responses**

#### **Success (200 OK)**

{

"success": true,

"blogDetails": {

"name": "Eye",

"description": "Blogs related to eyes",

"blogs": [],

"\_id": "6856a01391f9bf6c439ed295",

"\_\_v": 0

},

"message": "Category Created Successfully"

}

#### **Missing Fields (400 Bad Request)**

{

"success": false,

"message": "All fields are required"

}

#### **Server Error (500 Internal Server Error)**

{

"success": false,

"message": "Error message details"

}

**Authorization Required**

Yes , user should be Admin

**Notes**

* blogs array is empty on creation; blogs can later be associated with this category.
* Both name and description are mandatory for successful category creation.

**8. Show All Blog Categories**

### **Endpoint**

GET /api/v1/blogs/showAllCategories

### **Description**

Retrieves a list of all blog categories available in the system.

**Request**

#### **Headers**

* No special headers required

#### **Body**

* Not required for this GET request

**Responses**

#### **Success (200 OK)**

{

"success": true,

"data": [

{

"\_id": "6856a01391f9bf6c439ed295",

"name": "Eye",

"description": "Blogs related to eyes",

"blogs": [],

"\_\_v": 0

}

]

}

#### **Server Error (500 Internal Server Error)**

{

"success": false,

"message": "Error message details"

}

**Authentication Required**

No

### **Notes**

* Each blog category object includes its name, description, unique \_id, and an empty blogs array (until populated).
* Useful for frontends displaying category filters or for admin dashboards.