API Testing Documentation

Auth API

1.1 Register User

- **Endpoint:** http://localhost:8000/api/auth/register
- **Method:** POST
- **Description:** Register a new user. Required fields: name, phone, email, password, confirmPassword.
- **Headers:** None
- **Body:**


```
{
 "name": "Aniket Joshi",
 "phone": "1234567890",
 "email": "aniket@example.com",
 "password": "Password123!",
 "confirmPassword": "Password123!"
}
```


- **Response Example (201 Created):**


```
{
 "success": true,
 "data": {
  "user": {
   "_id": "<userId>",
   "name": "Aniket Joshi",
   "phone": "1234567890",
   "email": "aniket@example.com"
  },
  "tokens": {
   "accessToken": "...",
   "refreshToken": "..."
```

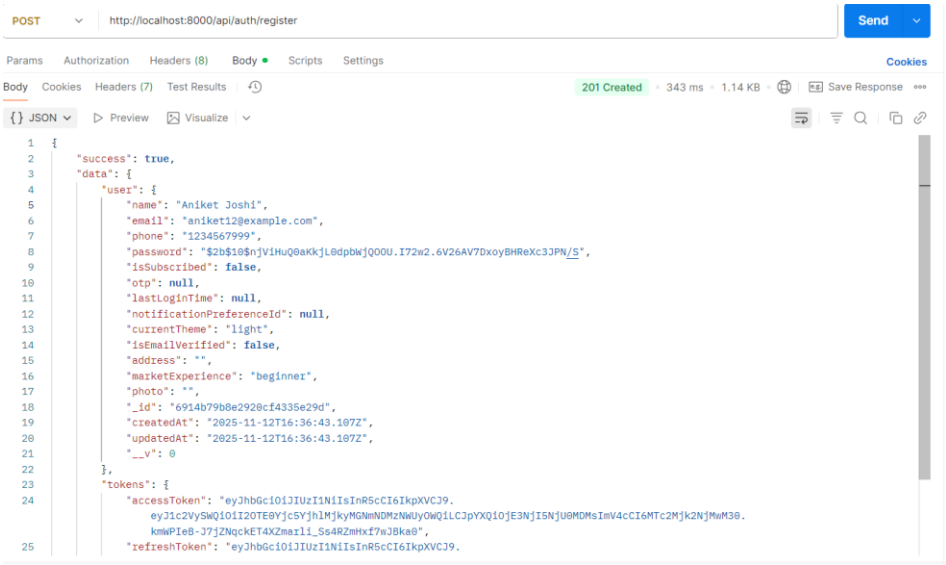
}

},

"message": "User registered successfully"

}

- **Screenshot:**



1.2 Login User

- **Endpoint:** http://localhost:8000/api/auth/login
- **Method:** POST
- **Description:** Login user with email and password to receive JWT tokens.
- **Headers:** None
- **Body:**

{

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

"password": "Password123!"

}

- **Response Example (200 OK):**

{

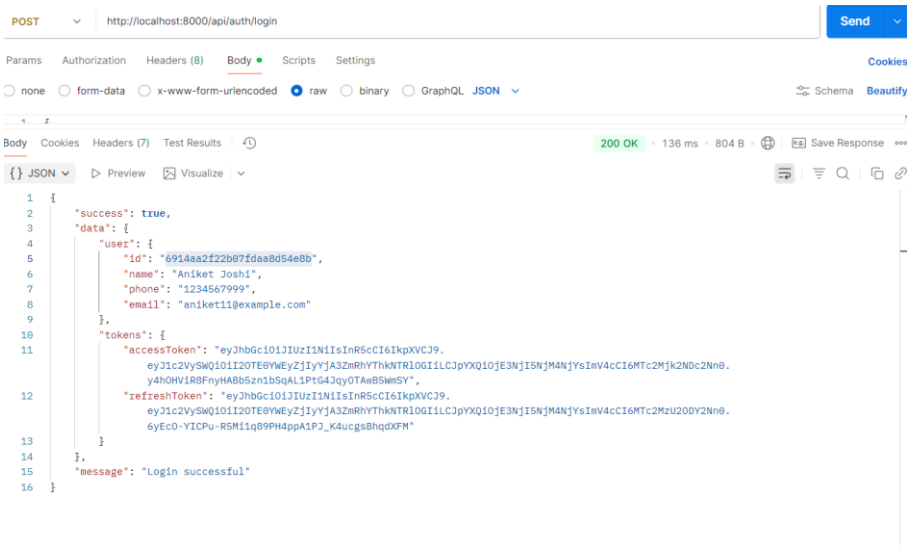
"success": true,

"data": {

"user": {

```
    "id": "<userId>",
    "name": "Aniket Joshi",
    "phone": "1234567890",
    "email": "aniket@example.com"
  },
  "tokens": {
    "accessToken": "...",
    "refreshToken": "..."
  }
 },
 "message": "Login successful"
}
```


- **Screenshot:**



1.3 Forgot Password

- **Endpoint:** http://localhost:8000/api/auth/forgot-password
- **Method:** POST
- **Description:** Generate OTP for password reset.
- **Headers:** None
- **Body:**


```
{
```

"email": "aniket@example.com"

}

- **Response Example (200 OK):**

{

"success": true,

"data": {

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

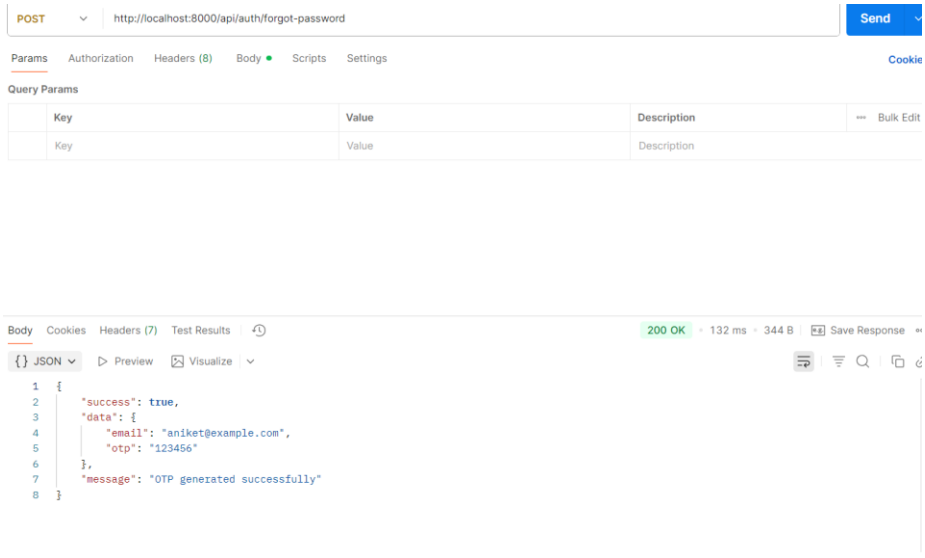
"otp": "123456"

},

"message": "OTP generated successfully"

}

- **Screenshot:**



1.4 Reset Password

- **Endpoint:** http://localhost:8000/api/auth/reset-password
- **Method:** POST
- **Description:** Reset password using OTP.
- **Headers:** None
- **Body:**

{

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

"otp": "123456",

"newPassword": "NewPass123!",

"confirmPassword": "NewPass123!"

}

- **Response Example (200 OK):**

{

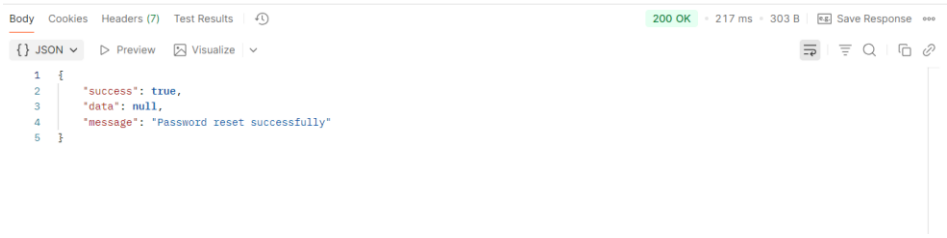
"success": true,

"data": null,

"message": "Password reset successfully"

}

- **Screenshot:**



1.5 Change Password

- **Endpoint:** http://localhost:8000/api/auth/change-password
- **Method:** POST
- **Description:** Change password for logged-in user.
- **Headers:**

Authorization: Bearer <accessToken>

- **Body:**

{

"oldPassword": "Password123!",

"newPassword": "NewPass123!",

"confirmPassword": "NewPass123!"

}

- **Response Example (200 OK):**

{

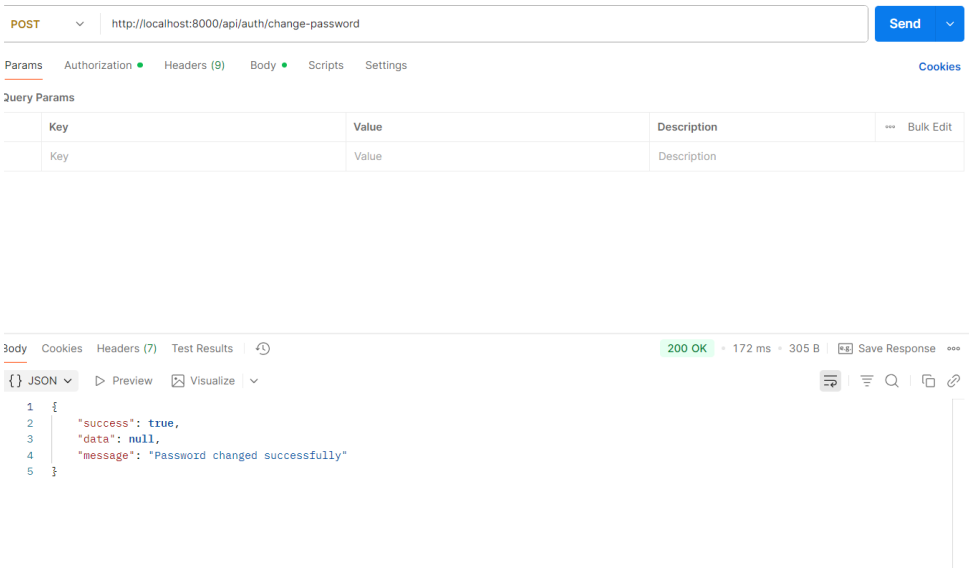
"success": true,

"data": null,

"message": "Password changed successfully"

}

- **Screenshot:**



1.6 Verify OTP

- **Endpoint:** http://localhost:8000/api/auth/verify-otp
- **Method:** POST
- **Description:** Verify OTP sent to email (temporary fixed OTP).
- **Headers:** None
- **Body:**

{

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

"otp": "123456"

}

- **Response Example (200 OK):**

{

"success": true,

"data": {

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

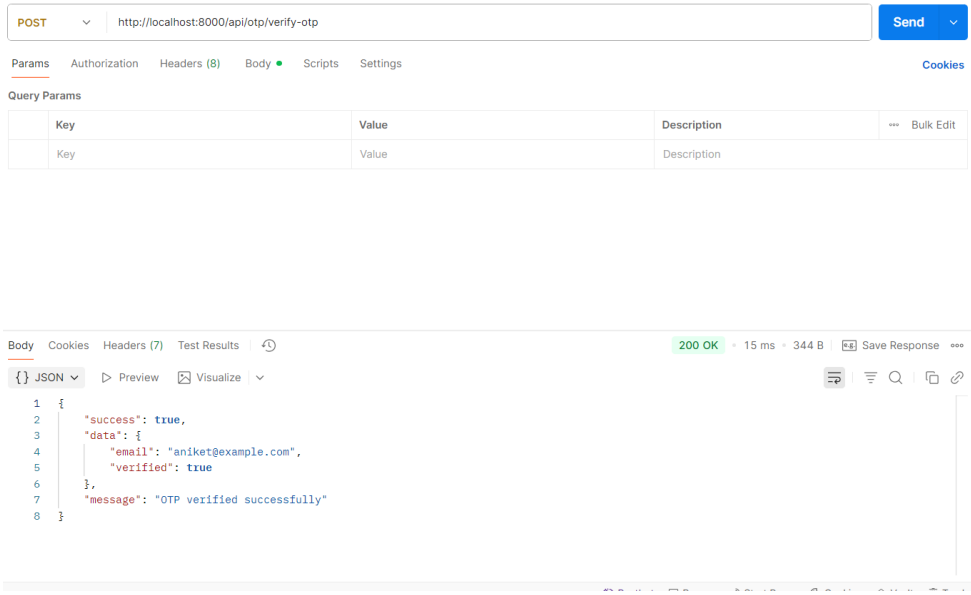
"verified": true

},

"message": "OTP verified successfully"

}

- **Screenshot:**



OTP API

2.1 Register OTP (Request/Verify)

- **Endpoint:** http://localhost:8000/api/otp/register
- **Method:** POST
- **Description:**
  - If only **email** is provided → request OTP (temporary fixed OTP 123456).
  - If **email + otp** is provided → verify OTP and return JWT tokens.
  - If email does not exist → temporary user is created.
- **Headers:** None
- **Request Body (Request OTP):**

{

"email": "aniket@example.com"

}

- **Response Example (200 OK - OTP Sent):**


```
{
  "success": true,
  "data": {
    "email": "aniket@example.com",
    "otp": "123456"
  },
  "message": "OTP sent successfully. (For testing OTP is 123456)"
}
```

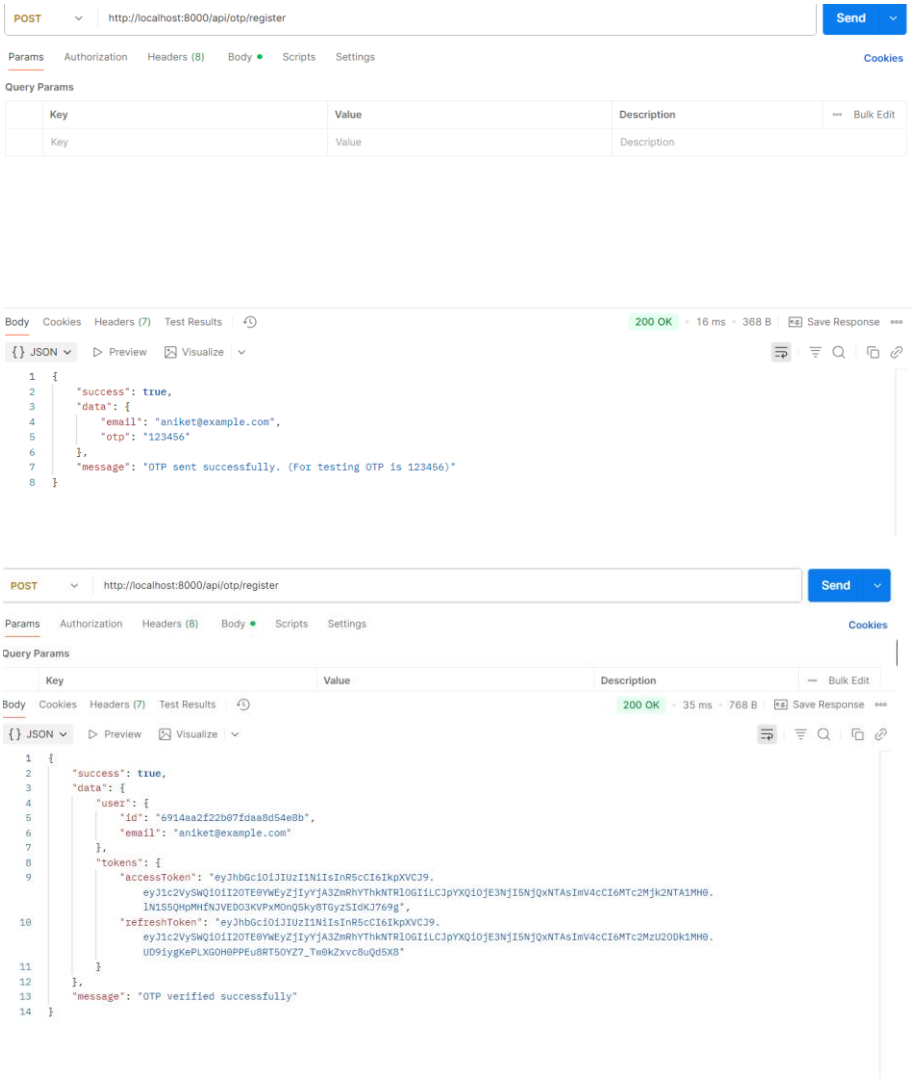

- **Request Body (Verify OTP):**


```
{
  "email": "aniket@example.com",
  "otp": "123456"
}
```


- **Response Example (200 OK - OTP Verified):**


```
{
  "success": true,
  "data": {
    "user": {
      "id": "<userId>",
      "email": "aniket@example.com"
    },
    "tokens": {
      "accessToken": "...",
      "refreshToken": "..."
    }
  },
  "message": "OTP verified successfully"
}
```

- **Screenshot:**



-

2.2 Login OTP (Request/Verify)

- **Endpoint:** http://localhost:8000/api/otp/login
- **Method:** POST
- **Description:**
    - If only **email** is provided → request OTP.
    - If **email + otp** is provided → verify OTP and return JWT tokens.
    - User must already exist.
- **Headers:** None
- **Request Body (Request OTP):**

{

```
  "email": "aniket@example.com"
}
```


- **Response Example (200 OK - OTP Sent):**


```
{
  "success": true,
  "data": {
    "email": "aniket@example.com",
    "otp": "123456"
  },
  "message": "Login OTP sent successfully. (For testing OTP is 123456)"
}
```

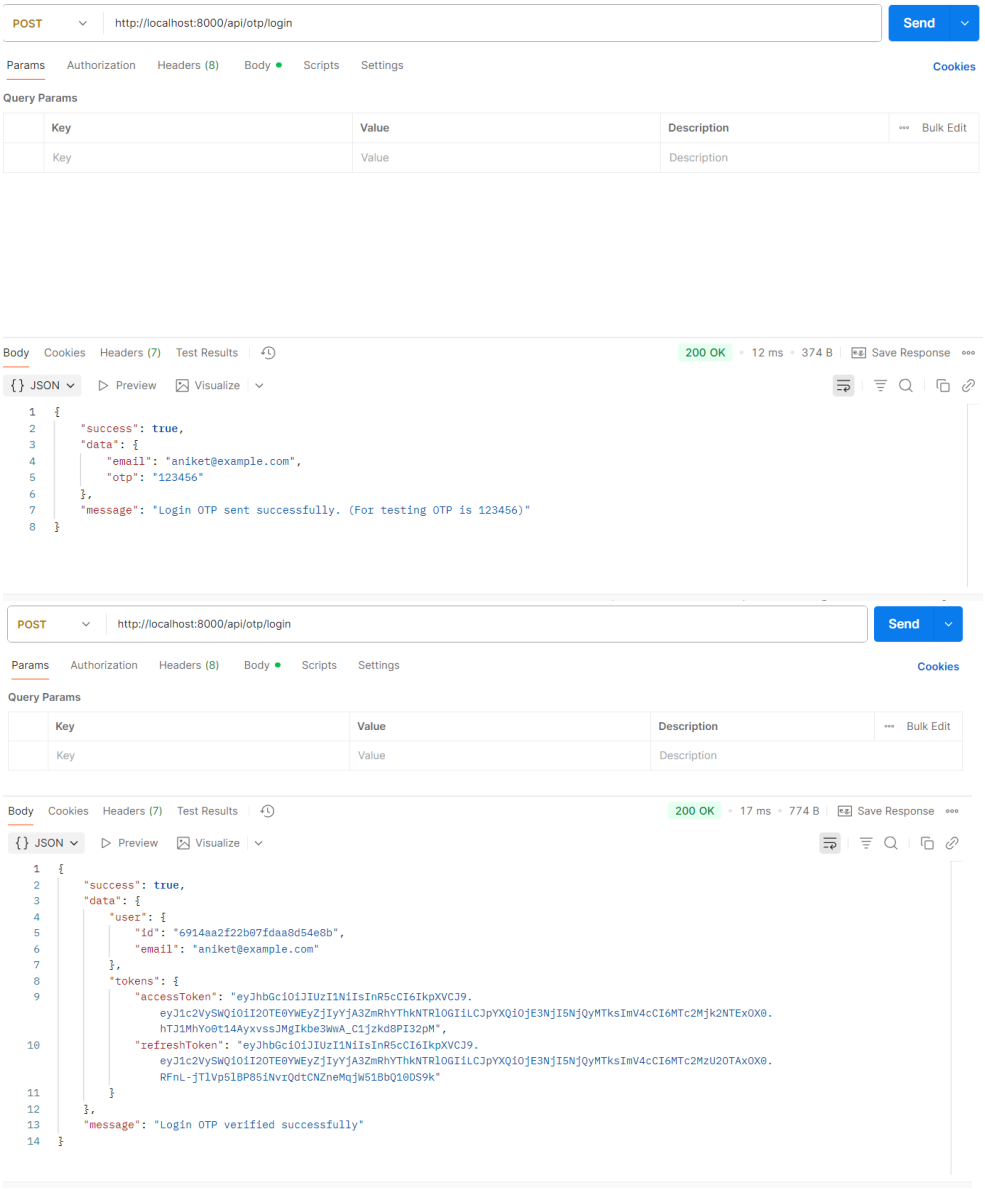

- **Request Body (Verify OTP):**


```
{
  "email": "aniket@example.com",
  "otp": "123456"
}
```


- **Response Example (200 OK - OTP Verified):**


```
{
  "success": true,
  "data": {
    "user": {
      "id": "<userId>",
      "email": "aniket@example.com"
    },
    "tokens": {
      "accessToken": "...",
      "refreshToken": "..."
    }
  },
  "message": "Login OTP verified successfully"
}
```

- **Screenshot:**



2.3 Verify OTP Directly

- **Endpoint:** http://localhost:8000/api/otp/verify-otp
- **Method:** POST
- **Description:** Verify OTP directly using email + OTP.
- **Headers:** None
- **Body:**

{

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

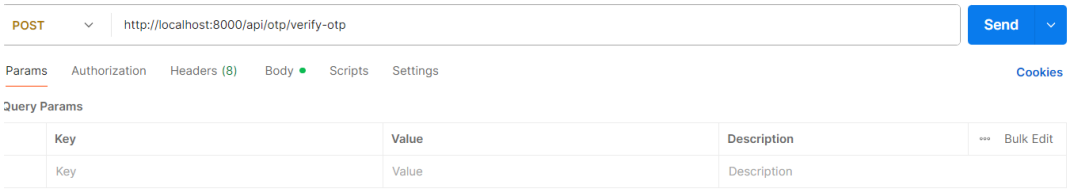
"otp": "123456"

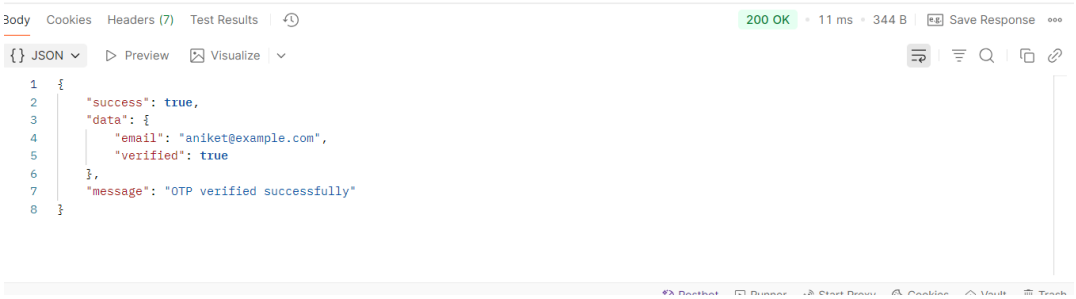
}

- **Response Example (200 OK):**


```
{
 "success": true,
 "data": {
  "email": "aniket@example.com",
  "verified": true
 },
 "message": "OTP verified successfully"
}
```


- **Screenshot:**





User API

3.1 Update User Profile

- **Endpoint:** http://localhost:8000/api/user/update/<userId>
- **Method:** POST
- **Description:** Update an existing user's profile. You can update **name, email, or phone**. At least one field should be provided.
- **Headers:**

Authorization: Bearer <accessToken>

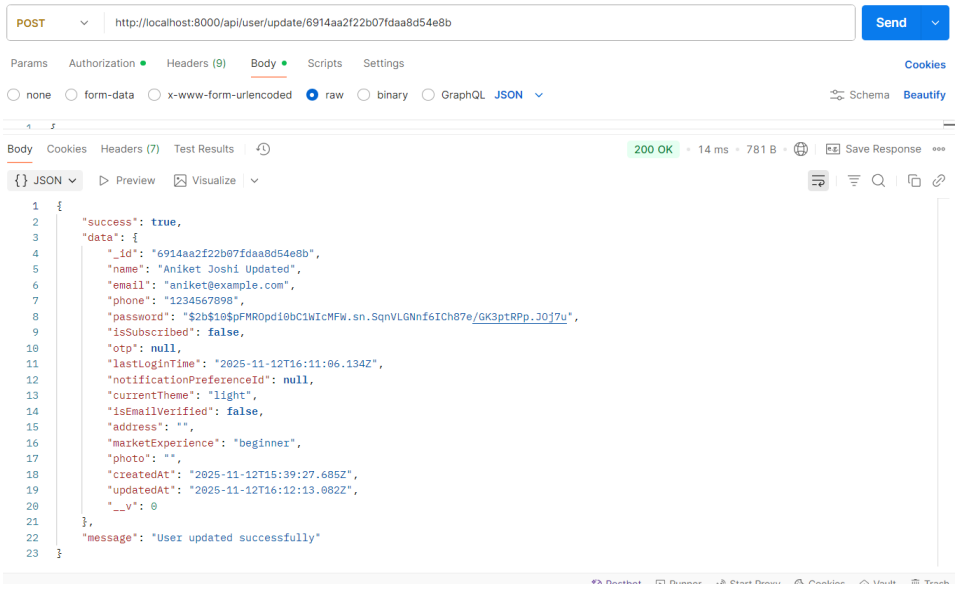
- **Request Body:**


```
{
 "name": "Aniket Joshi Updated",
 "email": "aniketupdated@example.com",
 "phone": "9876543210"
}
```


- **Response Example (200 OK):**


```
{
 "success": true,
 "data": {
  "_id": "<userId>",
  "name": "Aniket Joshi Updated",
  "email": "aniketupdated@example.com",
  "phone": "9876543210"
 },
 "message": "User updated successfully"
}
```


- **Screenshot:**



3.2 Delete User Profile

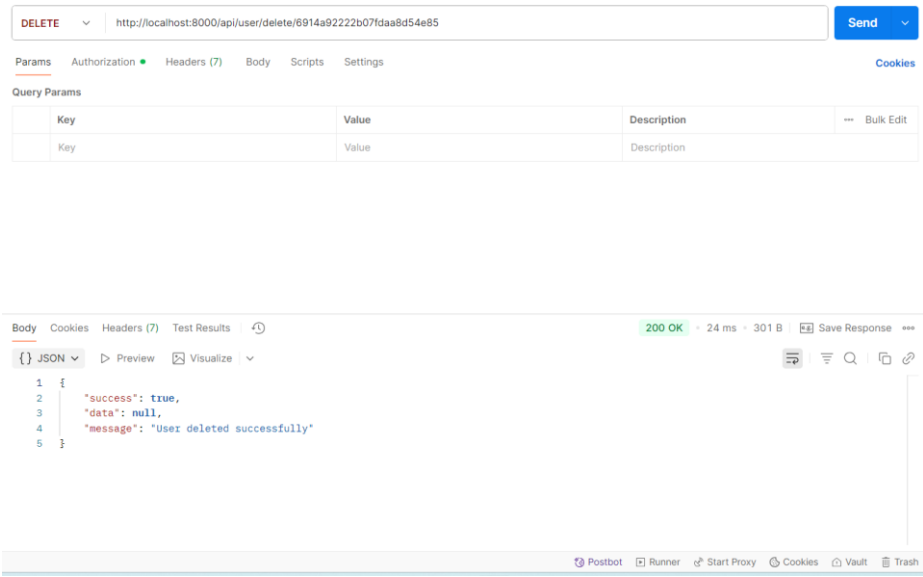
- **Endpoint:** http://localhost:8000/api/user/delete/<userId>
- **Method:** DELETE
- **Description:** Delete a user profile by ID. Only logged-in users can perform this action.
- **Headers:**

Authorization: Bearer <accessToken>

- **Request Body:** None
- **Response Example (200 OK):**


```
{
  "success": true,
  "data": null,
  "message": "User deleted successfully"
}
```


- **Screenshot:**



Notifications API

4.1 Create Notification

- **Endpoint:** http://localhost:8000/api/notifications
- **Method:** POST
- **Description:** Create a new notification. At least one of **photo, hyperlink, or description** is required.
- **Headers:**

Authorization: Bearer <accessToken>

- **Request Body:**

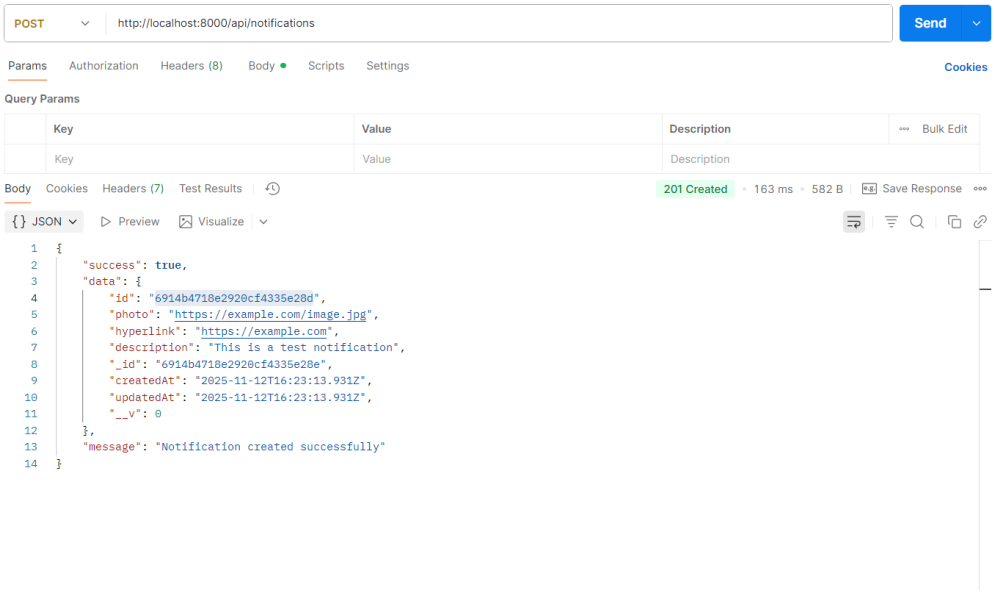

```
{
  "photo": "https://example.com/image.jpg",
  "hyperlink": "https://example.com",
  "description": "This is a test notification"
}
```


- **Response Example (201 Created):**


```
{
  "success": true,
  "data": {
    "_id": "<notificationId>",
    "photo": "https://example.com/image.jpg",
    "hyperlink": "https://example.com",
    "description": "This is a test notification",
    "createdAt": "2025-11-12T10:00:00.000Z",
    "updatedAt": "2025-11-12T10:00:00.000Z"
  },
  "message": "Notification created successfully"
```

}

- **Screenshot:**



4.2 Get All Notifications

- **Endpoint:** http://localhost:8000/api/notifications
- **Method:** GET
- **Description:** Fetch all notifications sorted by latest first.
- **Headers:** Optional (if your app requires auth)
- **Request Body:** None
- **Response Example (200 OK):**

{

"success": true,

"data": [

{

"_id": "<notificationId>",

"photo": "https://example.com/image.jpg",

"hyperlink": "https://example.com",

"description": "This is a test notification",

"createdAt": "2025-11-12T10:00:00.000Z",

"updatedAt": "2025-11-12T10:00:00.000Z"

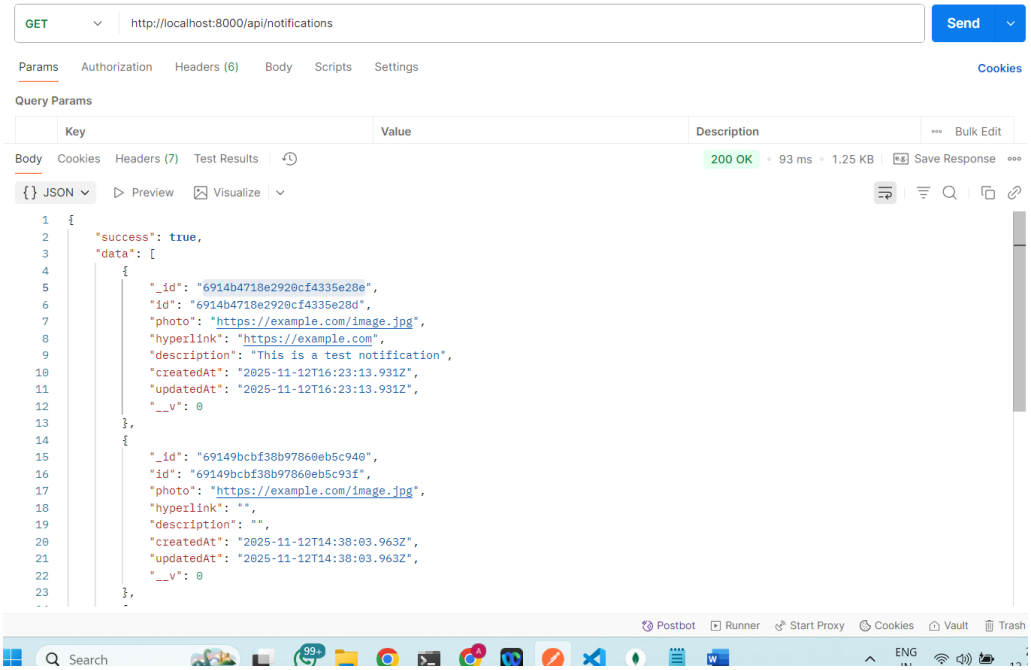
}

],

"message": "Notifications fetched successfully"

}

- **Screenshot:**



4.3 Update Notification

- **Endpoint:** http://localhost:8000/api/notifications/update/<notificationId>
- **Method:** POST
- **Description:** Update a notification. At least one of **photo, hyperlink, or description** must be provided.
- **Headers:**

Authorization: Bearer <accessToken>

- **Request Body:**

{

"description": "Updated notification description"

}

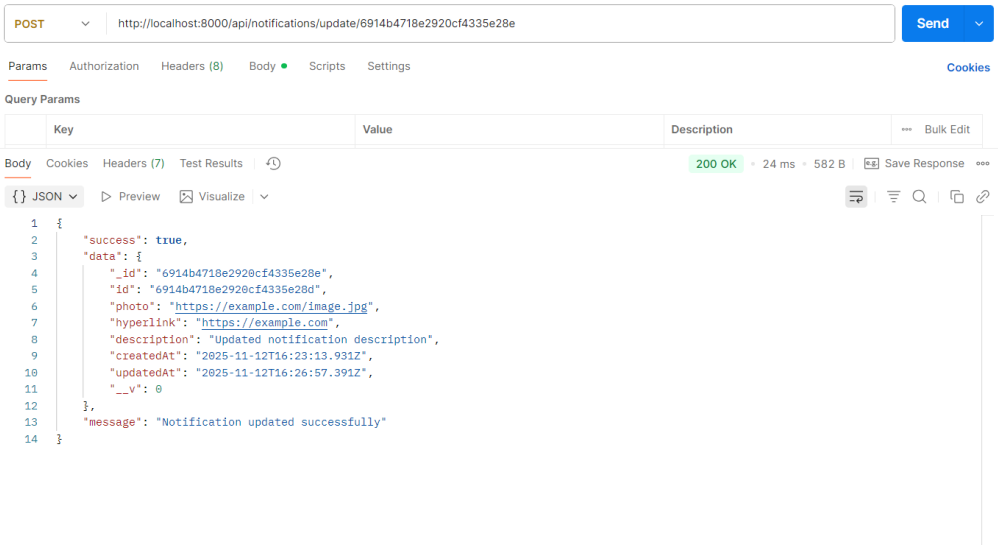
- **Response Example (200 OK):**

{

"success": true,

```
 "data": {
   "_id": "<notificationId>",
   "photo": "https://example.com/image.jpg",
   "hyperlink": "https://example.com",
   "description": "Updated notification description",
   "createdAt": "2025-11-12T10:00:00.000Z",
   "updatedAt": "2025-11-12T12:00:00.000Z"
 },
 "message": "Notification updated successfully"
}
```


- **Screenshot:**



4.4 Delete Notification

- **Endpoint:** http://localhost:8000/api/notifications/<notificationId>
- **Method:** DELETE
- **Description:** Delete a notification by ID.
- **Headers:**

Authorization: Bearer <accessToken>

- **Request Body:** None
- **Response Example (200 OK):**


```
{
```

"success": true,

"data": null,

"message": "Notification deleted successfully"

}

- **Screenshot:**